NETWORK ALIGNMENT

Dr. Alioune Ngom
School of Computer Science
University of Windsor
angom@uwindsor.ca

Winter 2013

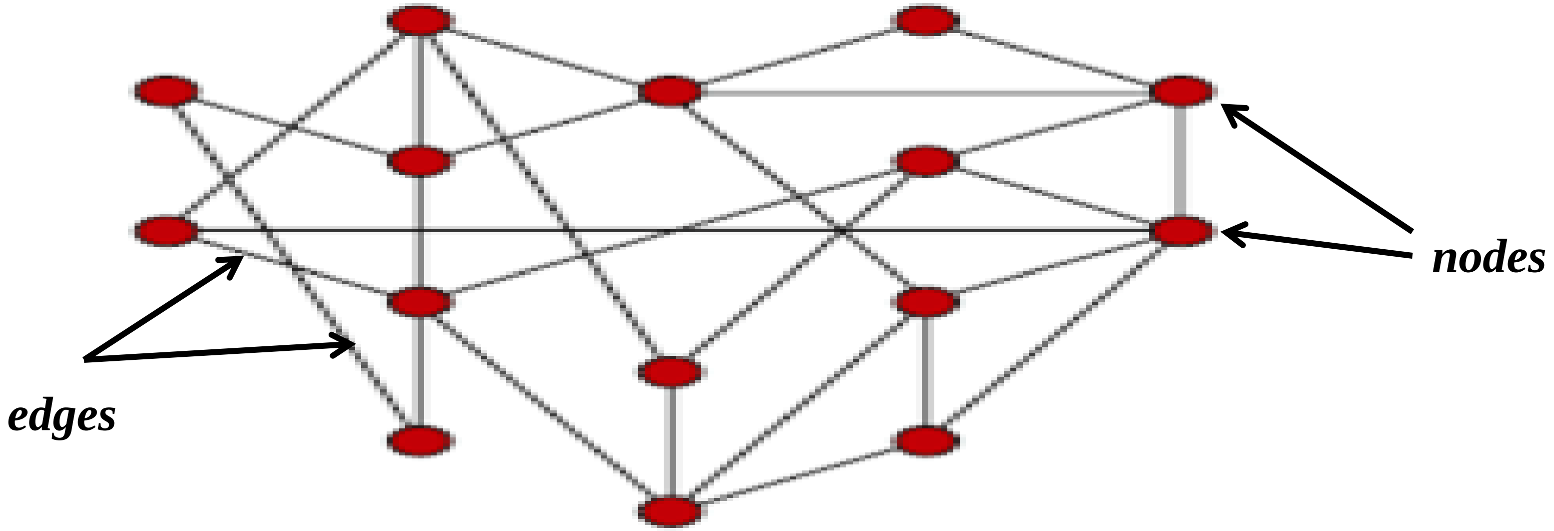
What are Proteins ?

- Proteins are large biological molecules consisting of one or more chains of amino acids.
- Proteins perform vast array of functions within living organisms. For example:
  - Transporting molecules from one location to another.
  - Responding to stimuli
  - Replicating DNA
  - Catalyzing metabolic reactions

- The study of protein interactions is fundamental in understanding how proteins function.

Protein-Protein Interaction Network

- **Protein–protein interactions** occur when two or more proteins bind together, often to carry out their biological functions.
- A graphical representation of protein-protein interaction is known as **Protein-Protein Interaction (PPI) network.**
- In a PPI network, all proteins are represented as nodes and interactions as edges between the nodes.



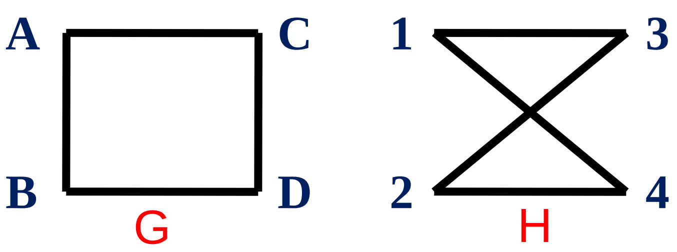
Challenges in PPI Network

- Network comparison is the process of contrasting two or more protein interaction networks, representing different species.
  - It helps to understand the structure, function and evolution of proteins in different species.

- The **problem** in comparing/aligning networks is lack of knowledge of how each node of one network maps to one or more nodes of the other networks.
- Absence of this information requires solving the sub-graph isomorphism problem.

Challenges in PPI Network

- *Sub-graph isomorphism:* An isomorphism is a bijection between nodes of two networks G and H that preserves edge adjacency.

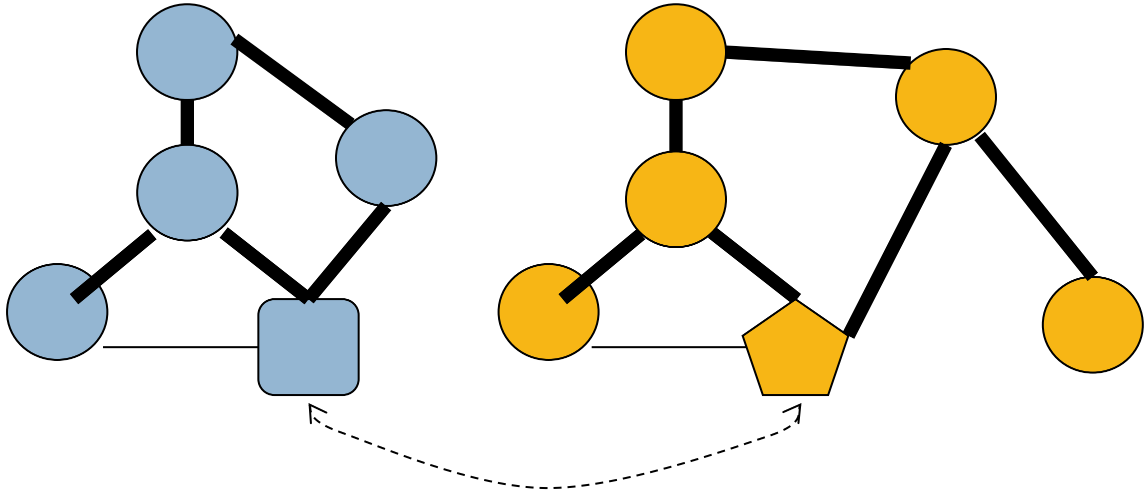


A→1
B→2
C→3
D→4

- Exact comparisons are inappropriate in biology (biological variation)
- Network alignment
  - More general problem is finding the best way to "fit" G into H even if G and H do not have exact sub-graph
  - Thus, an efficient and accurate multiple network alignment algorithm is required.

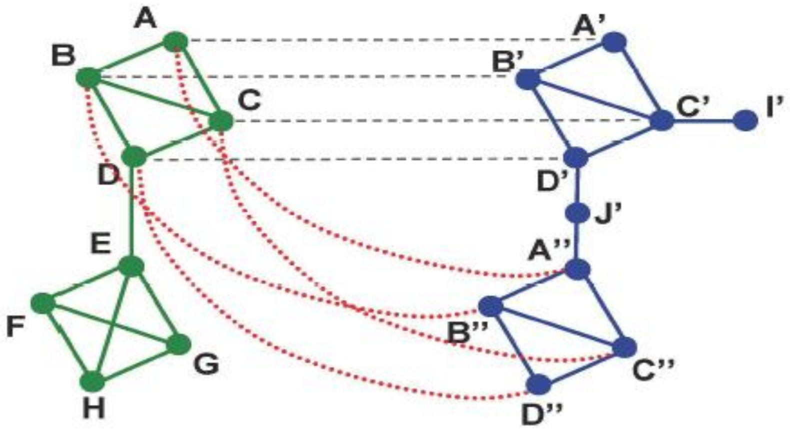
Network Alignment

- The process of overall comparison is commonly applied to detect sub-networks that are conserved across species and thus represent true functional modules.
- **“Conserved”** means two sub graphs contain proteins serving **similar** functions, having **similar** interaction profiles, etc.
  - Key word is similar, not identical



Network Alignment

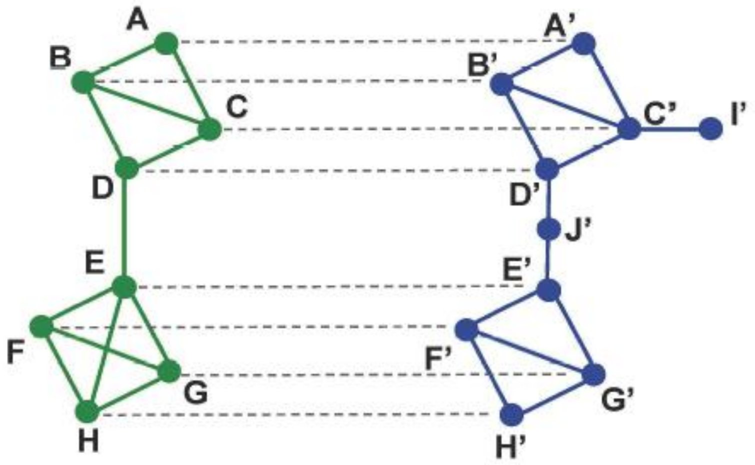
- □ Various methods of alignment:
  - ➢ **Global vs. Local**
  - ➢ Pairwise vs. Multiple
  - ➢ Functional vs. Topological



- □ Local Alignment:
  - ▪ Mappings are chosen independently for each region of similarity
  - ▪ Can be ambiguous, with one node having pairings in different local alignments
  - ▪ Example algorithms:
    - ▪ ***PathBLAST, NetworkBLAST, MaWISh, Graemlin***

Network Alignment

- □ Methods vary in these aspects:
  - ➢ **Global vs. Local**
  - ➢ Pairwise vs. Multiple
  - ➢ Functional vs. Topological



- □ **Global Alignment**:
  - ▪ Provides best overall alignment from every node in one network to nodes in the other network.
  - ▪ May lead to sub-optimal matchings in some local regions
  - ▪ Example algorithms:
    - ▪ ***GRAAL, IsoRank, IsoRankN, Extended Graemlin***

Network Alignment

- Methods vary in these aspects:
  - Global vs. Local
  - **Pairwise vs. Multiple**
  - Functional vs. Topological
- **Pairwise Alignment**:
  - Two networks aligned
  - Example algorithms:
    - ***GRAAL, PathBLAST, MaWISh, IsoRank***
- **Multiple Alignment**:
  - More than two networks aligned
  - Example algorithms:
    - ***Greamlin, Extended PathBLAST, Extended IsoRank***

Network Alignment

- Methods vary in these aspects:
  - Global vs. Local
  - Pairwise vs. Multiple
  - **Functional vs. Topological**

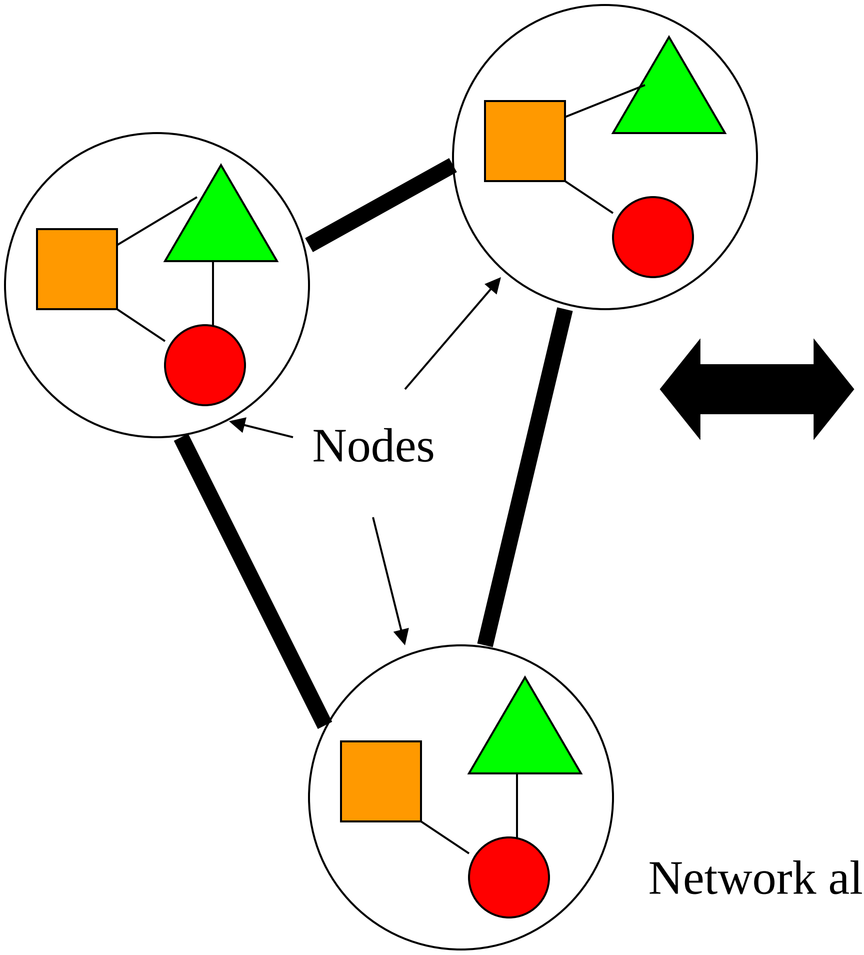
- **Functional Information**
  - Information external to network topology used, e.g., protein sequence, to define "similarity" between nodes
  - Example algorithms:
    - all except for *GRAAL;* e.g. *IsoRank*, but then perform poorly
- **Topological Information**
  - Only network topology used to define node "similarity".
  - It is interesting, as it answers how much and what type of

Network Alignment Graph

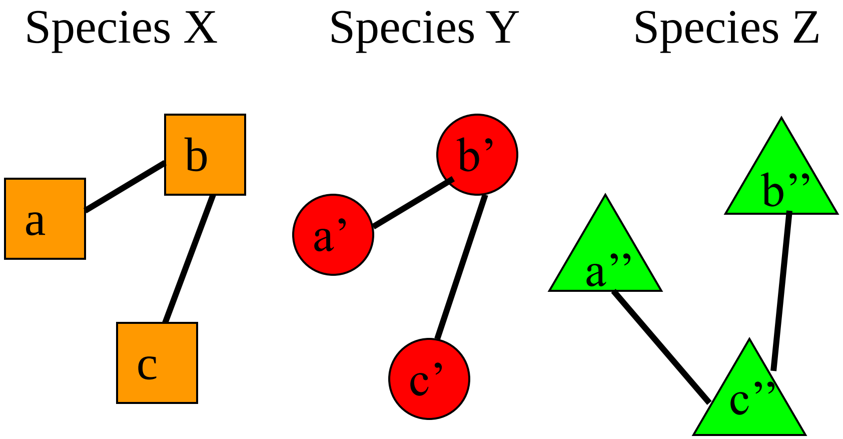
- **One heuristic approach:**
- A merged representation of the networks being compared is created, called a "***network alignment graph***" in which:
  - *Nodes* represent sets of proteins, one from each network
  - *Edges* represent conserved protein interactions across different networks
  - The alignment is simple when there exists a 1-to-1 correspondence between proteins across the networks, but in general there may be a many-to-many correspondence

- Then apply a greedy algorithm for identifying conserved sub-networks embedded in the "network alignment graph"

Network Alignment Graph- Example



Network alignment graph



Individual species’ PPI network

Assignment Problem

- An assignment problem seeks to minimize/maximize the total cost assignment of '*N*' rows to '*M*' columns of a given matrix c_{ij}.
- ***Total Cost Function :***

$$\sum_{i=1}^{N}\sum_{j=1}^{M} c_{ij}x_{ij} \rightarrow \max$$

$$\sum_{j=1}^{N} x_{ij} = 1 \qquad \forall i \in 1.....N$$

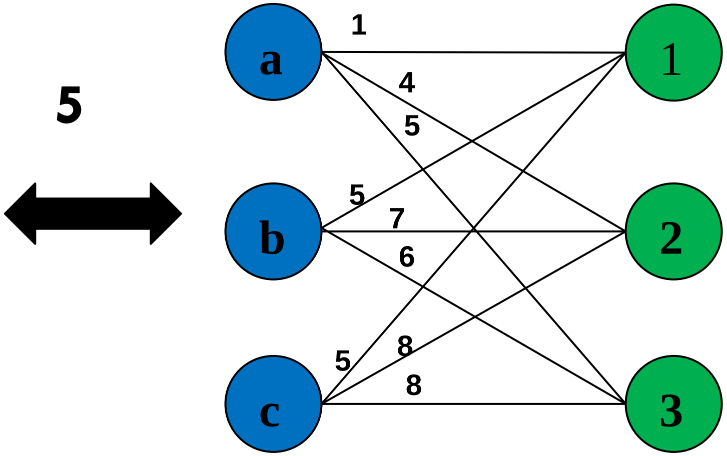
$$\sum_{i=1}^{N} x_{ij} = 1 \qquad \forall j \in 1.....M$$

- $\{c_{ij}\}_{\mathrm{NxN}}$ - cost matrix, where c_{ij} - cost of assigning an element in row ***i*** to an element in column ***j***.
- $\{x_{ij}\}_{\mathrm{NxN}}$ - resulting binary matrix, where x_{ij} = 1 if and only if an element of ***i***th row is assigned to an element in ***j***th job.

Hungarian Algorithm

- The **Hungarian method** is an algorithm which solves the assignment problem [*Kuhn et. al, 1979 and 2005*].
- Special considerations include:
  - If the number of rows does not equal the number of columns add dummy rows/columns with 0 assignment costs as needed.
- Network representation in terms of a bipartite graph.

$$\begin{pmatrix} 1 & 4 & 5 \\ 5 & 7 & 6 \\ 5 & 8 & 8 \end{pmatrix}$$



Hungarian Algorithm-Contd..

- ***Maximization Problem***

District

Salesman

$$\begin{array}{c|ccccc} & \mathbf{A} & \mathbf{B} & \mathbf{C} & \mathbf{D} & \mathbf{E} \\ \hline \mathbf{1} & 32 & 38 & 40 & 28 & 40 \\ \mathbf{2} & 40 & 24 & 28 & 21 & 36 \\ \mathbf{3} & 41 & 27 & 33 & 30 & 37 \\ \mathbf{4} & 22 & 38 & 41 & 36 & 36 \\ \mathbf{5} & 29 & 33 & 40 & 35 & 39 \end{array}$$

Problem : *Find the assignment of salesmen to districts that will result in maximum sales.*

Hungarian Algorithm-Contd..

- ***Conversion to Minimization Problem*** **-** The given maximization problem is converted into minimization problem by subtracting from the highest sales value (i.e., 41) with all elements of the given table.

District

Salesman

$$\begin{array}{cc} & \begin{array}{ccccc} \mathbf{A} & \mathbf{B} & \mathbf{C} & \mathbf{D} & \mathbf{E} \end{array} \\ \begin{array}{c} \mathbf{1} \\ \mathbf{2} \\ \mathbf{3} \\ \mathbf{4} \\ \mathbf{5} \end{array} & \begin{pmatrix} 9 & 3 & 1 & 13 & 1 \\ 1 & 17 & 13 & 20 & 5 \\ 0 & 14 & 8 & 11 & 4 \\ 19 & 3 & 0 & 5 & 5 \\ 12 & 8 & 1 & 6 & 2 \end{pmatrix} \end{array}$$

Hungarian Algorithm-Contd..

- *Step 1:* ***Matrix Reduced Row-wise***

District

Salesman

$$\begin{array}{c|ccccc} & \mathbf{A} & \mathbf{B} & \mathbf{C} & \mathbf{D} & \mathbf{E} \\ \hline \mathbf{1} & 8 & 2 & 0 & 12 & 0 \\ \mathbf{2} & 0 & 18 & 12 & 19 & 4 \\ \mathbf{3} & 0 & 14 & 8 & 11 & 4 \\ \mathbf{4} & 19 & 3 & 0 & 5 & 5 \\ \mathbf{5} & 11 & 7 & 0 & 5 & 1 \end{array}$$

Hungarian Algorithm-Contd..

- ***Step 2: Matrix Reduced Column-wise and Zeros Covered :*** Reduce the matrix column-wise and draw minimum number of lines to cover all the zeros in the matrix, as shown below.

$$\begin{array}{lc} & \textbf{\textit{District}} \\ & \begin{array}{ccccc} \mathbf{A} & \mathbf{B} & \mathbf{C} & \mathbf{D} & \mathbf{E} \end{array} \\ \textbf{\textit{Salesman}} \begin{array}{c} \mathbf{1} \\ \mathbf{2} \\ \mathbf{3} \\ \mathbf{4} \\ \mathbf{5} \end{array} & \left(\begin{array}{ccccc} 8 & 0 & 0 & 7 & 0 \\ 0 & 14 & 12 & 14 & 4 \\ 0 & 12 & 8 & 6 & 4 \\ 19 & 1 & 0 & 0 & 5 \\ 11 & 7 & 0 & 0 & 1 \end{array} \right) \end{array}$$

Hungarian Algorithm-Contd…

- ***Step 3: Add & Subtract the least Uncovered Element:*** Number of lines drawn ≠ Order of matrix. Hence not optimal.
- Select the least uncovered element, i.e., 4 and subtract it from other uncovered elements, add it to the elements at intersection of line and leave the elements that are covered with single line unchanged

$$\begin{array}{llcccccc}
 & & & & \textbf{\textit{District}} & & \\
 & & \textbf{A} & \textbf{B} & \textbf{C} & \textbf{D} & \textbf{E} \\
 & \textbf{1} & 12 & 0 & 0 & 7 & 0 \\
 & \textbf{2} & 0 & 10 & 8 & 10 & 0 \\
\textbf{\textit{Salesman}} & \textbf{3} & 0 & 8 & 4 & 2 & 0 \\
 & \textbf{4} & 23 & 1 & 0 & 0 & 5 \\
 & \textbf{5} & 15 & 5 & 0 & 0 & 1
\end{array}$$

Hungarian Algorithm-Contd..

- ***Step 4: Final Assignments -*** Now, number of lines drawn = Order of matrix, hence optimality is reached.

District

Salesman

$$
\begin{array}{c|ccccc}
 & \textbf{A} & \textbf{B} & \textbf{C} & \textbf{D} & \textbf{E} \\
\hline
\textbf{1} & 12 & \boxed{0} & \xcancel{0} & 7 & \xcancel{0} \\
\textbf{2} & \boxed{0} & 10 & 8 & 10 & \xcancel{0} \\
\textbf{3} & \xcancel{0} & 8 & 4 & 2 & \boxed{0} \\
\textbf{4} & 23 & 1 & \boxed{0} & \xcancel{0} & 5 \\
\textbf{5} & 15 & 5 & \xcancel{0} & \boxed{0} & 1
\end{array}
$$

Motivation

- The availability of huge quantity of data on protein interactions has motivated researchers to compare the networks of different species.
- The alignment of bio-molecular networks is used for examining interactions in the networks of different species.
- Thus network alignment allows us to-
  - Identify conserved functional modules
  - Predict protein functions
  - Validate protein interactions
  - Predict protein interactions
  - Discovering protein complexes
- The primary focus of my thesis is on multiple alignment PPI network.

Related Work

Pairwise Alignment

- Various methods have been proposed for aligning two protein-protein interaction networks

- For example – *GRAAL*, *PathBlast*, *NetworkBlast*,*Pinalog*.

- PINALOG is an example of pairwise alignment which forms the alignment between two PPINs based on the similarities of protein sequence and the protein function similarity between the two networks. [*Phan H.T.T. et. al. 2012*]

- It is a global alignment method comprises of three main steps:
  - (i) **Community detection:** identifies dense sub-networks of input networks using Cfinder.

Pairwise Alignment – Contd..

- (ii) **Community mapping:** maps similar communities that have high similarity scores (Hungarian Method).

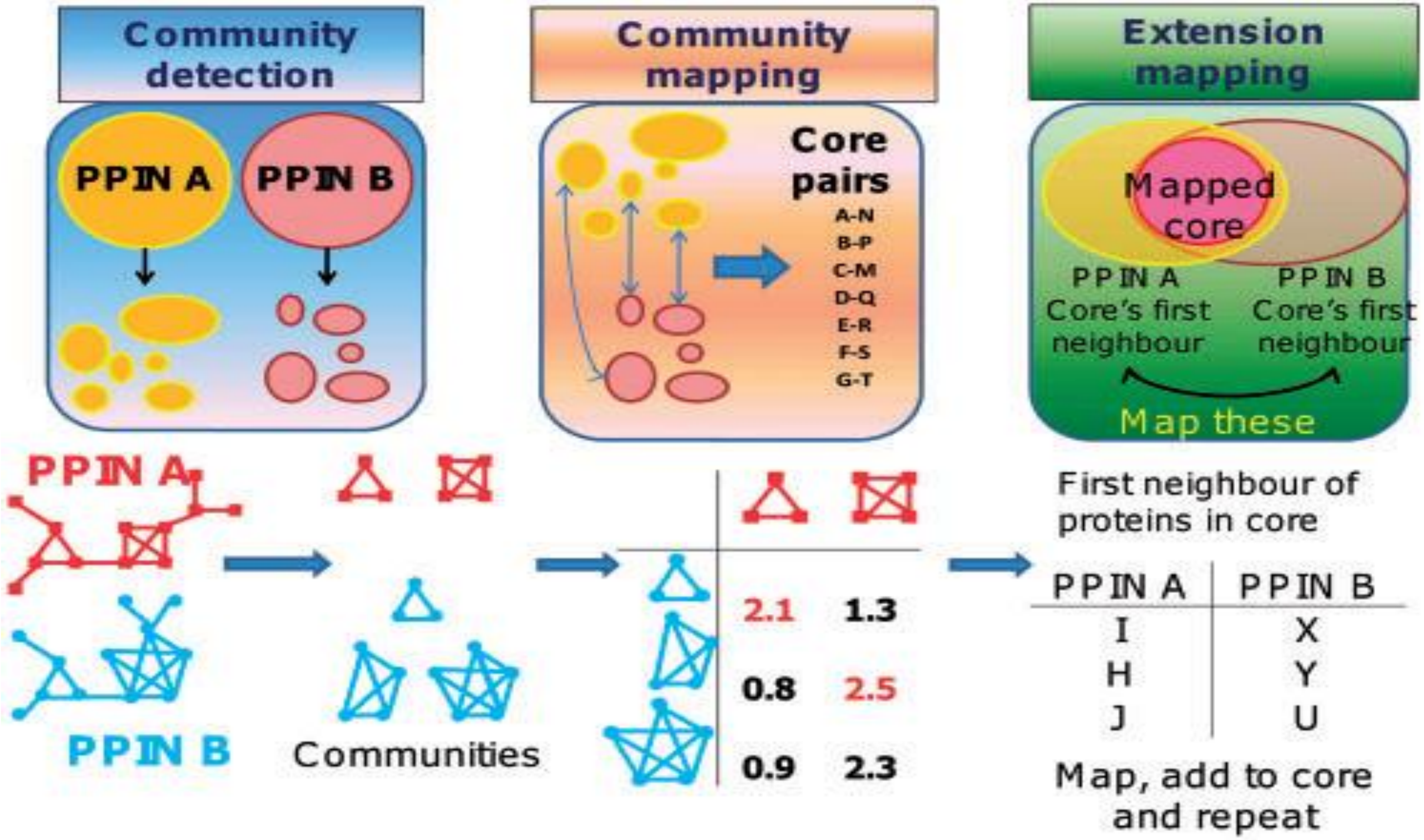
$$F(\,C_i{}^A, C_j{}^B) = \sum_{\substack{a_k \in C_i{}^A \\ b_l \in C_j{}^B \\ a_k, bl \in OptMap}} s(ak, bl)$$

  - Similar protein pairs from mapped communities are extracted to form a list of core pairs.

$$F(core) = \sum_{\substack{C_i{}^A \subset A \\ C_j{}^B \subset B}} F(\,C_i{}^A,\, C_j{}^B)$$

- (iii) **Extension mapping:** maps proteins in the neighbourhood of the core protein pairs which are then added to the core. This step is repeated until no more pair is added.

Pairwise Alignment – Contd..



2.1	1.3
0.8	2.5
0.9	2.3

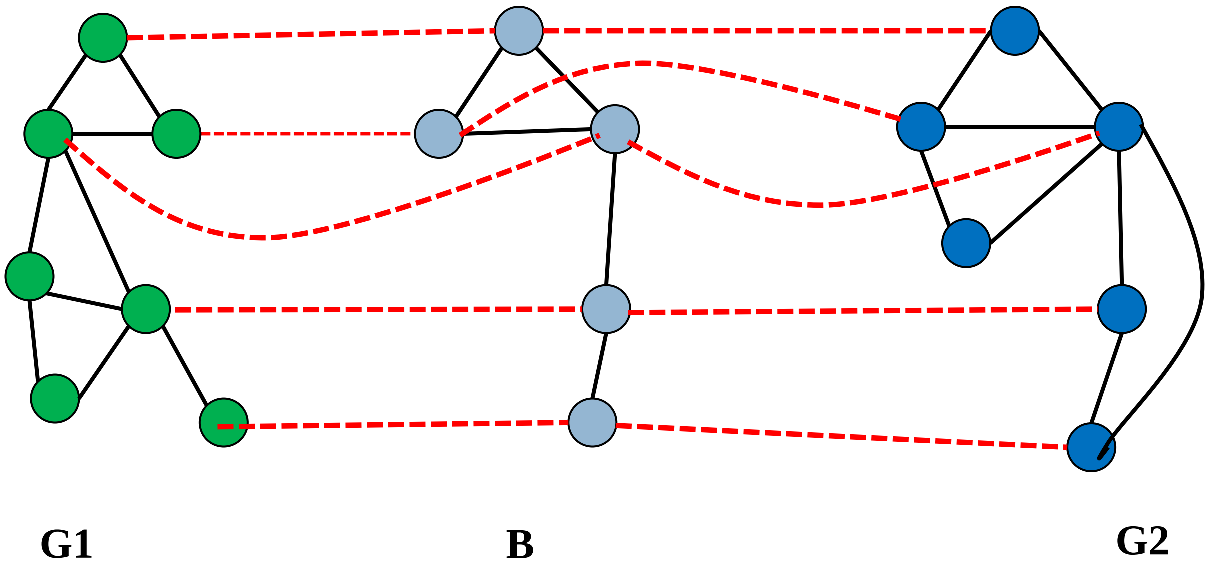
PPIN A	PPIN B
I	X
H	Y
J	U

Multiple Alignment

- Multiple alignment means comparing more than two networks.
- Every alignment method generally consist of two parts:
  - Maximizing the size of common subgraph between the input networks
  - Including the information(sequence/functional similarity) to the alignment.
- Various algorithms have been proposed for aligning multiple PPI networks.
- Example : *Graemlin, IsoRank/IsoRankN.*
- Explanation of IsoRank [*Singh,R. et al. (2008)*] method for aligning multiple networks:-
- Given *k* PPI networks, first the similarity scores of every pair of cross-species proteins is computed

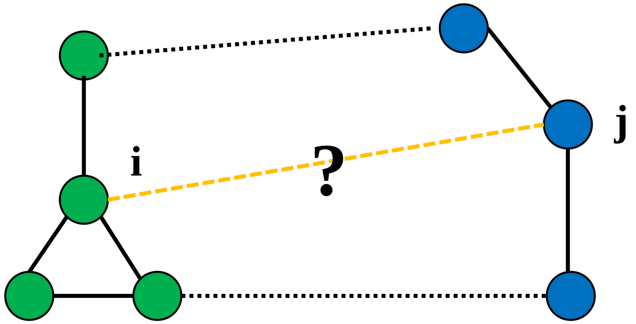
Multiple Alignment-IsoRank

- □ **Input –** weighted Graphs G1 and G2 with weights between 0 and 1.
- □ **Output**
  - □ Maximum Common Subgraph – largest subgraph B that is isomorphic to subgraph of G1 and G2.



Multiple Alignment-IsoRank

- □ Two nodes are a good match if their respective neighbours also match well.



- □ To check the quality of mapping ***R****ij* function is defined

$$\boldsymbol{R_{ij}} = \sum_{u \epsilon N(i)} \sum_{v \epsilon N(j)} \frac{i}{|N(u)||N(v)|} R_{uv}$$

- N(u) and N(v) are neighbors of node i and j.

The formula calculates overall pairings in between the neighbours of i and j. After calculating ***R****ij* find the matching between the graphs.

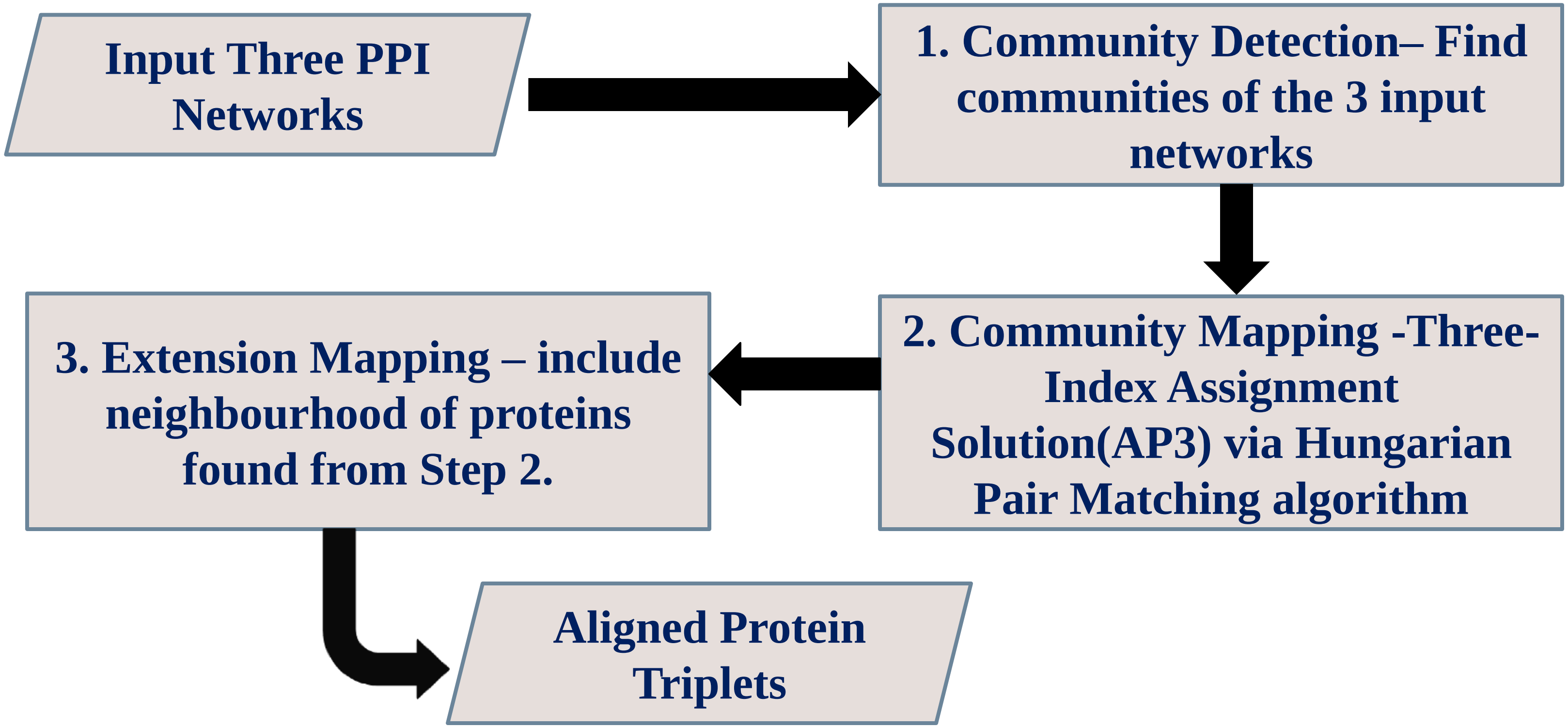
Multiple Alignment-IsoRank

- Thus, the score of a protein pair depends on the score of their neighbours, which in turn, depend on the neighbours of their neighbours, and so on.
- Once these ‘topological’ scores are computed for all node pairs, sequence-based BLAST scores are included in the alignment scores.
- ISORANK then constructs the node alignment with the repetitive greedy strategy to identify all proteins with highest scores
- It then outputs those proteins, and removing all scores involving any of the identified nodes.

Proposed Method

Proposed Method

- The proposed method is an extension of PINALOG from pairwise alignment method to aligning three protein interaction networks.



Input Three PPI Networks

1. Community Detection– Find communities of the 3 input networks

2. Community Mapping -Three-Index Assignment Solution(AP3) via Hungarian Pair Matching algorithm

3. Extension Mapping – include neighbourhood of proteins found from Step 2.

Aligned Protein Triplets

Community Detection

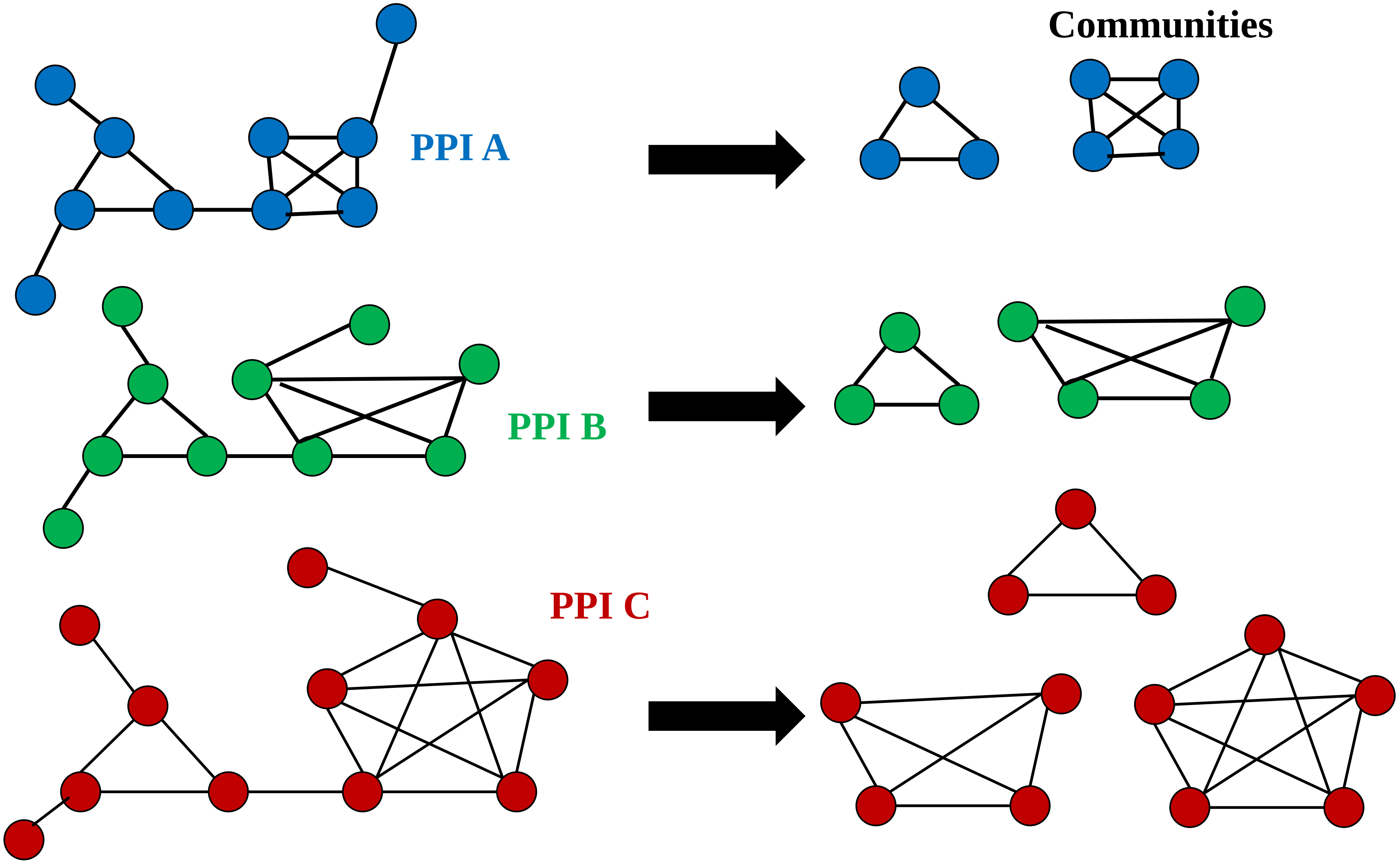
□ Protein complexes or functional modules form highly connected components in the PPI networks.

□ It is better to find highly connected components(clusters) of the PPI networks first and to align them, rather than aligning the networks directly. *(Brohee et. al, [2006])*.

□ Various clustering methods have been proposed for finding the clusters of protein interaction networks.

□ Clustering method used here is Cfinder.(k=3)

Community Detection



Node Scoring Scheme

- The sequence similarity of two proteins a_i and b_j is calculated based on their BLAST bit score as:

$$s_{seq}(ai, bj) = \frac{S(ai, bj)}{\sqrt{S(a_i, ai)S(bj, bj)}}$$

where $S(a_i,b_j)$ is the BLAST bit score value when aligning a_i and b_j.

- The similarity between nodes of the networks is a combination of protein sequence s_{seq} and functional similarity s_f

Community Mapping

- This step involves matching the communities obtained from the previous step having high similarity scores.
- One of the Three-Index Assignment Solution is used to find complete match between the three networks.
  - Using Hungarian algorithm.

- The AP3 is an optimization problem on a complete tripartite graph
- The cost of choosing triangle $(i, j, k)\ is\ \boldsymbol{c_{ijk}}$
- The objective of AP3 is to choose "N" disjoint triangles *(i, j, k)* so that the total cost is maximized.

Community Mapping-Contd..

Three-Index Assignment

Total Cost Function :

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N} c_{ijk}x_{ijk} \rightarrow max$$

$$\sum_{j=1}^{N} x_{ijk} = 1 \text{ or } 0$$

$$\forall i , j, k \epsilon 1 \ldots\ldots N$$

Hungarian(Pair) Assignment

Total Cost Function :

$$\sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}x_{ij} \rightarrow max$$

$$\sum_{j=1}^{N} x_{ij} = 1 \qquad \forall i \epsilon 1 \ldots\ldots N$$

$$\sum_{i=1}^{N} x_{ij} = 1 \qquad \forall j \epsilon 1 \ldots\ldots N$$

Community Mapping-Contd..

- AP3 consists of two permutations (say p and q), while a solution to AP2 consists of only one permutation (say q).
- Solution - optimize one permutation subject to the other permutation being fixed.

$$\mathbf{max} \sum_{j=1}^{N} c_{i,p(i),q(i)}$$

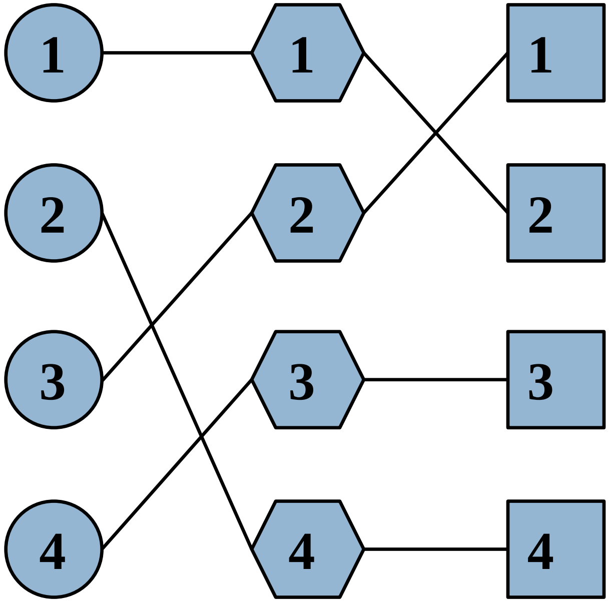
- Here we fix permutation ***p*** and optimize permutation ***q.****(becomes AP2 problem)*
- Thus the value of $d_{i,j}$ is calculated as follows:

$$d_{i,j} = c_{i,j} + c_{j,k}$$

Same approach is used for other steps as well.

Community Mapping-Contd..

- Given three graphs; first we consider a random initial assignment.



1, 2, 3, 4(index)

p = (1, 4, 2, 3)

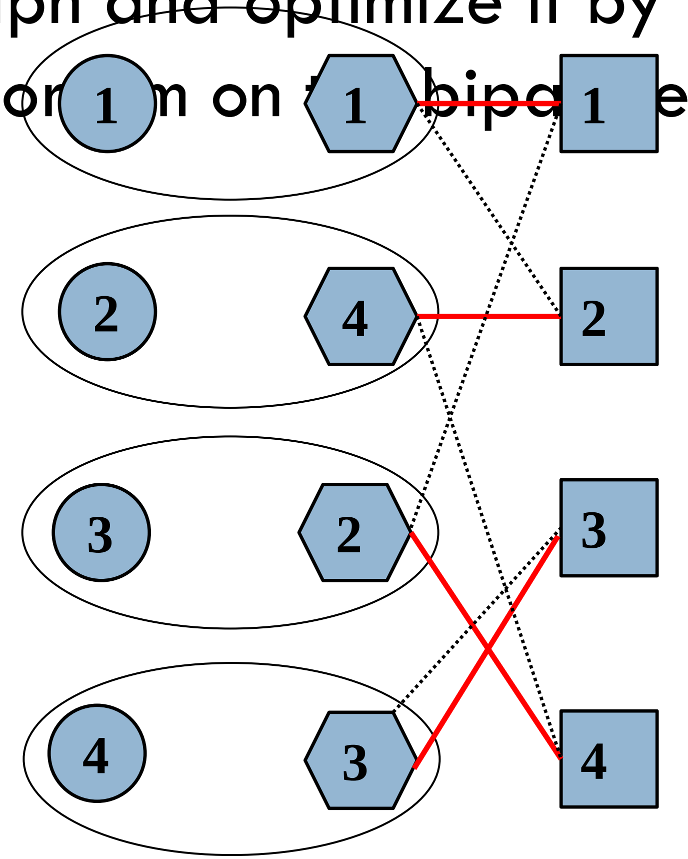
q = (2, 4, 1, 3)

Say total cost = 17

- ***p*** is matching between graph 1 and 2 and ***q*** between 1 and 3.

Community Mapping-Contd..

Optimize permutation q- Construct corresponding bipartite graph and optimize it by applying the Hungarian Algorithm on the bipartite graph.



Initial assignment

1, 2, 3, 4(index)

p = (1, 4, 2, 3)

q = (2, 4, 1, 3)

Cost = *17*

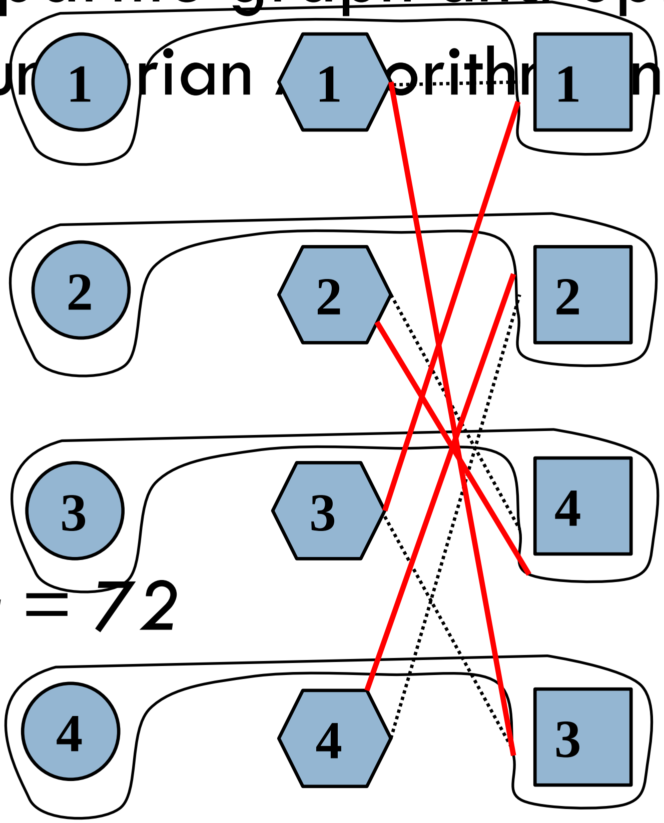
New assignment

1, 2, 3, 4(index)

p = (1, 4, 2, 3)

Community Mapping-Contd..

- ***Optimize permutation*** **p-** Construct corresponding bipartite graph and optimize it by applying the Hungarian Algorithm on the bipartite graph.



Cost = 72

Updated assignment

1, 2, 3, 4(index)

p = (1, 4, 2, 3)

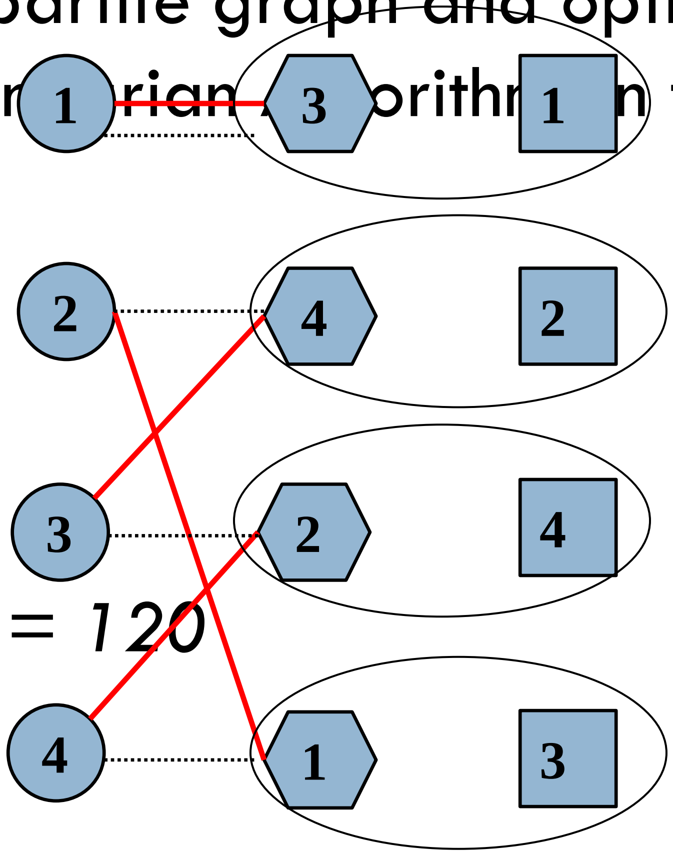
q = (1, 2, 4, 3)

New assignment

1, 2, 3, 4(index)

Community Mapping-Contd..

- *Optimize index* **permutation-** Construct corresponding bipartite graph and optimize it by applying the Hungarian Algorithm on the bipartite graph.



Cost = 120

Updated assignment

1, 2, 3, 4(index)

p = (1, 4, 2, 3)

q = (1, 2, 4, 3)

New assignment

1, 3, 4, 2(*)(index)

Extension Mapping

- In addition to protein sequence similarity, topological similarity is added.
- The protein triplets obtained after second step are considered as core protein triplets.
- Neighbours of proteins in the core are considered as candidates for extension mapping.
- Let $N(a_i)$ and $N(b_i)$ be the set of all first neighbours (proteins separated by one interaction) and second neighbours (proteins separated by two interactions) of a_i in A and b_i in B

Extension Mapping-Contd..

□ The similarity between a_i and b_j in extension mapping is then defined as $\boldsymbol{s_{ext}(a_i, bj)}$

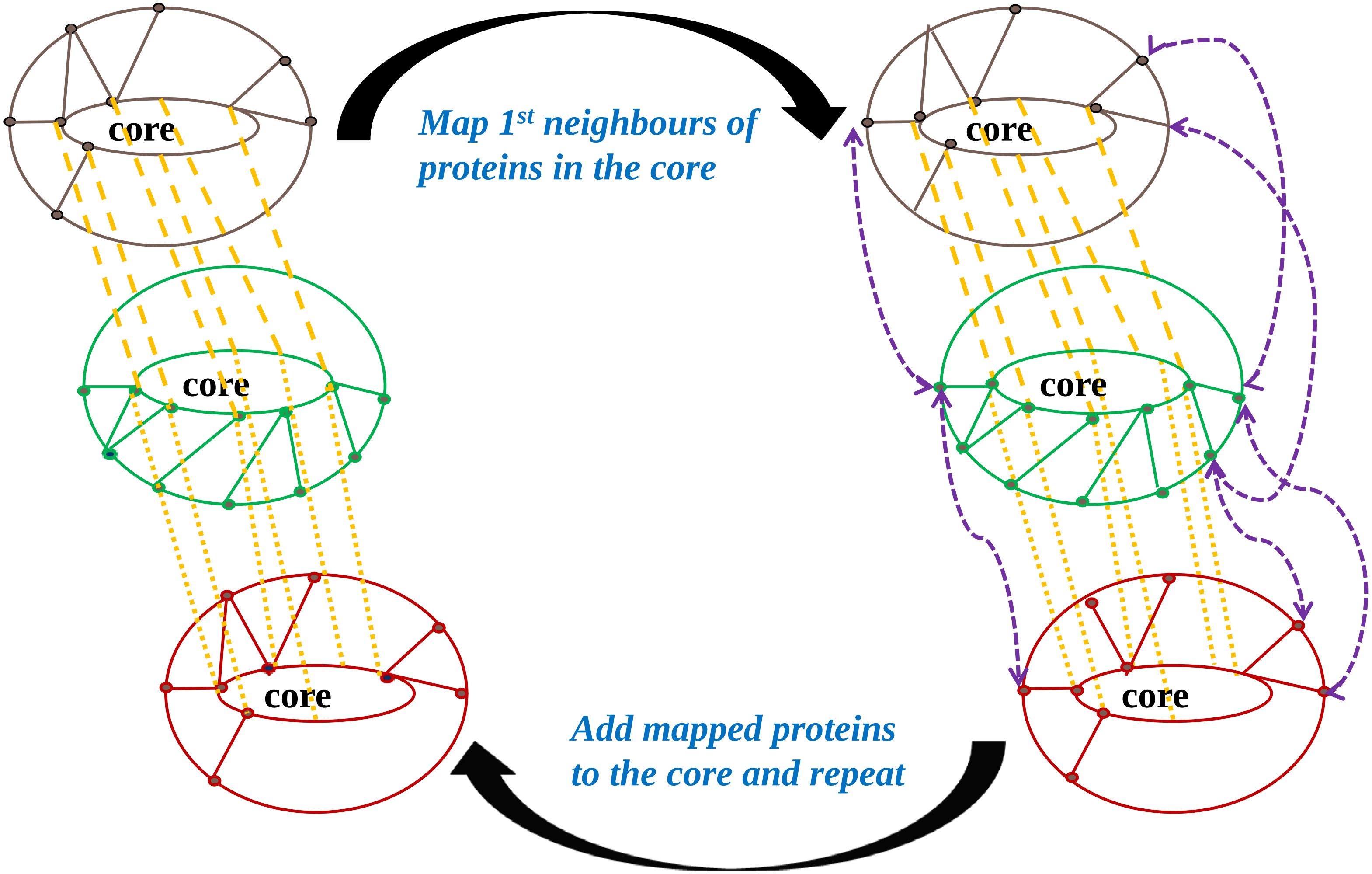
$$\boldsymbol{s(a_i, bj) + \sum_{\substack{a_k \in N(ai) \\ b_l \in N(bj) \\ a_k, bl \in core}} \frac{1}{((d(a_k, ai) + 1)(d(b_l, bj) + 1))} s(ak, bl)}$$

where $d(a_k, a_i)$ refers to the distance between the nodes

□ Candidates are mapped where the scores of candidate protein pairs include a part of the similarities of their aligned neighbours in the core.

□ This step is performed until no more proteins can be added in the core.

Extension Mapping-Contd..



Map 1st neighbours of proteins in the core

Add mapped proteins to the core and repeat

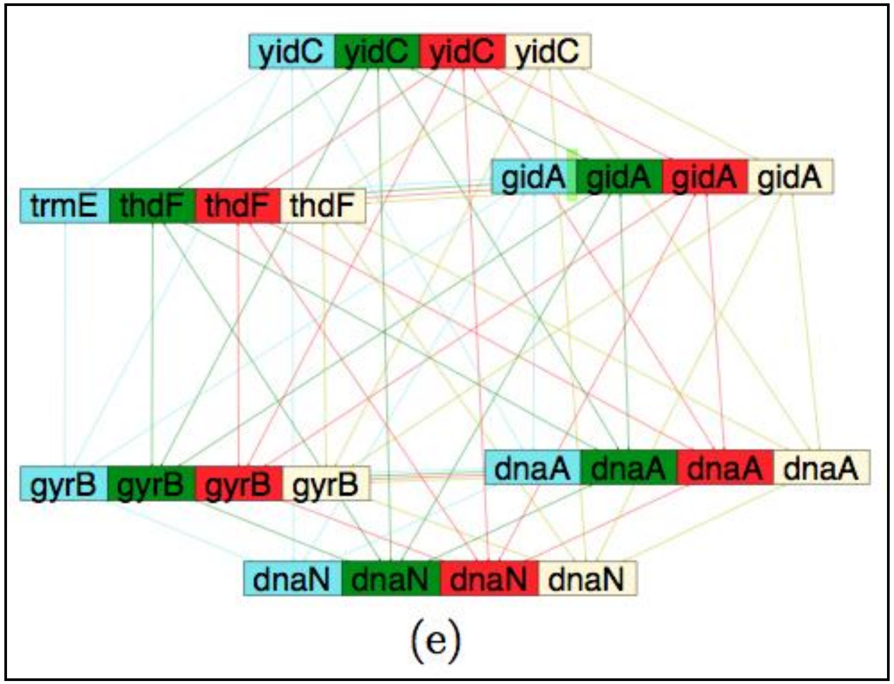
Introduction

- Sequence alignment seeks to identify conserved DNA or protein sequence
  - Intuition: conservation implies functionality

EFTPPVQAAYQKVVAGV	(human)
DFNPNVQAAFQKVVAGV	(pig)
EFTPPVQAAYQKVVAGV	(rabbit)

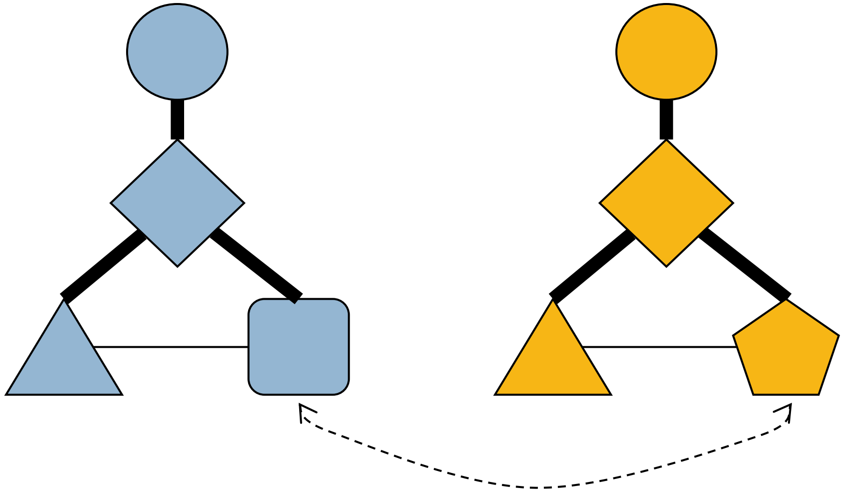
Introduction

- By similar intuition, subnetworks conserved across species are likely functional modules



Introduction

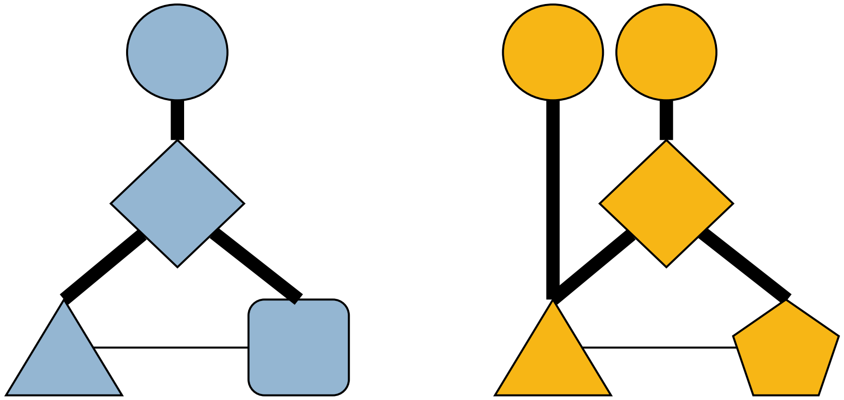
- "Conserved" means two subgraphs contain proteins serving **similar** functions, having **similar** interaction profiles
  - Key word is similar, not identical



mismatch/substitution

Introduction

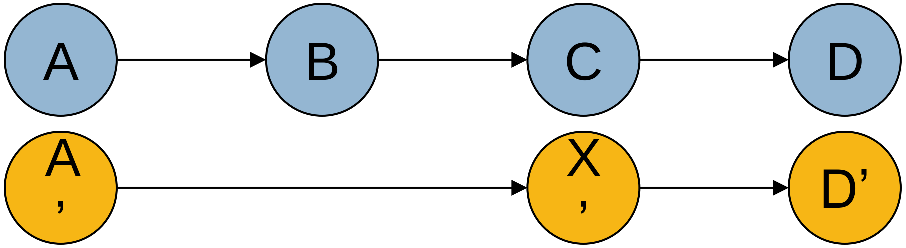
- **Inter**actions conserved in ortho**logs**
  - Orthology is a fuzzy notion
  - Sequence similarity not necessary for conservation of function



Introduction

Early approaches: PathBLAST

- Goal: identify conserved *pathways* (chains)
- Idea: can be done efficiently by dynamic programming (DP) if networks are DAGs



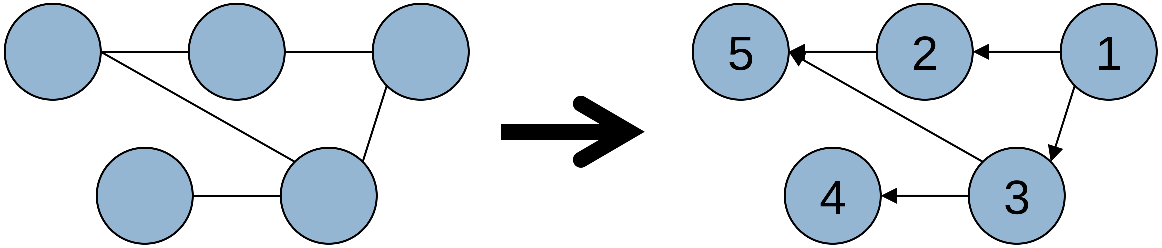
Score: match + gap + mismatch + match

Kelley et al (2003)

Introduction

Early approaches: PathBLAST

- Problem: Networks are neither acyclic nor directed
- Solution: eliminate cycles by imposing random ordering on nodes, perform DP; repeat many times



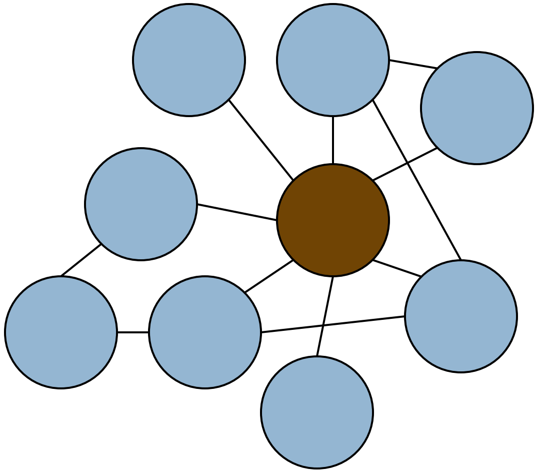
- In expectation, finds conserved paths of length *L* within networks of size *n* in O(*L*!*n*) time
- Drawbacks
  - Computationally expensive
  - Restricts search to specific topology

Kelley et al (2003)

Introduction

Early approaches: MaWISh

- Goal: identify conserved *multi-protein complexes* (clique-like structures)
- Idea: such structures will likely contain at least one *hub* (high-degree node)

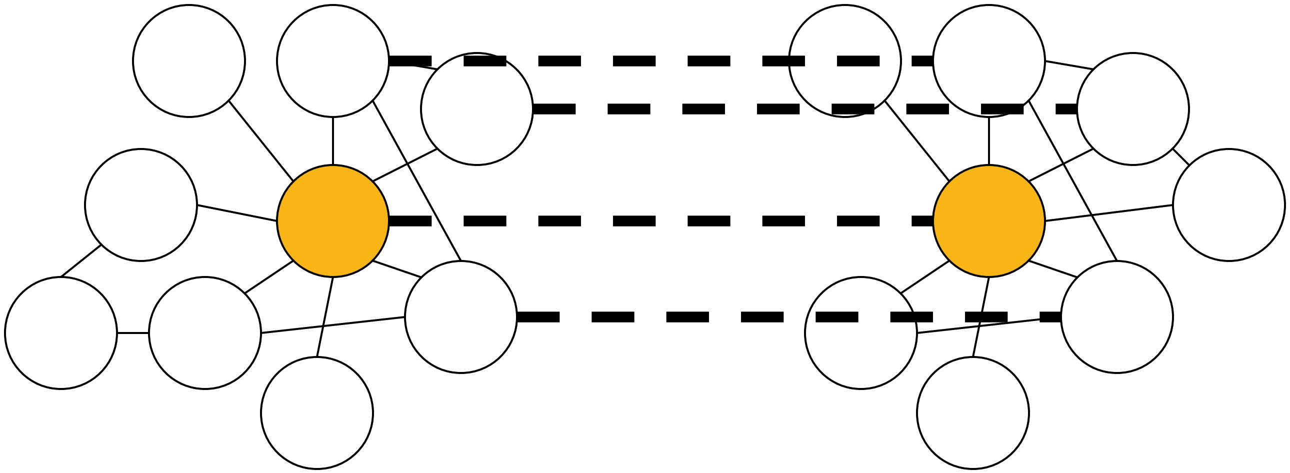


Koyuturk et al (2004)

Introduction

Early approaches: MaWISh

- Algorithm: start by aligning a pair of homologous hubs, extend greedily



Efficient running time, but also only solves a specific case (specific topology, here cliques)

Koyuturk et al (2004)

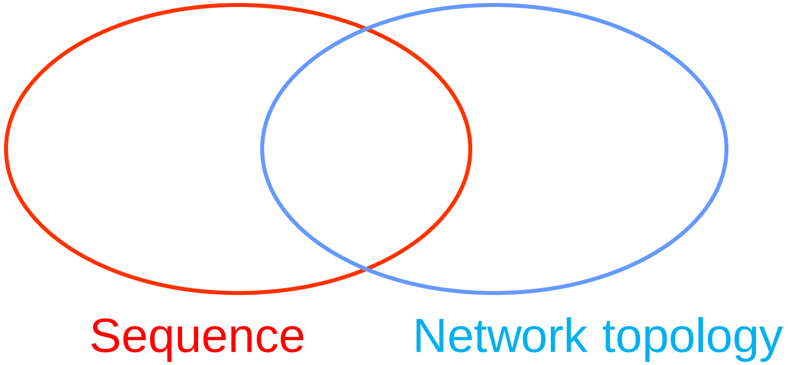
Introduction

A General Network Aligner: Goals

- Solve restrictions of existing approaches
  - Should extend gracefully to multiple alignment
    - PathBLAST was extended to 3-way alignment, but extension scales exponentially in number of species
  - Should not restrict search to specific network topologies (cliques/pathways)
- Must be efficient in running time

Introduction

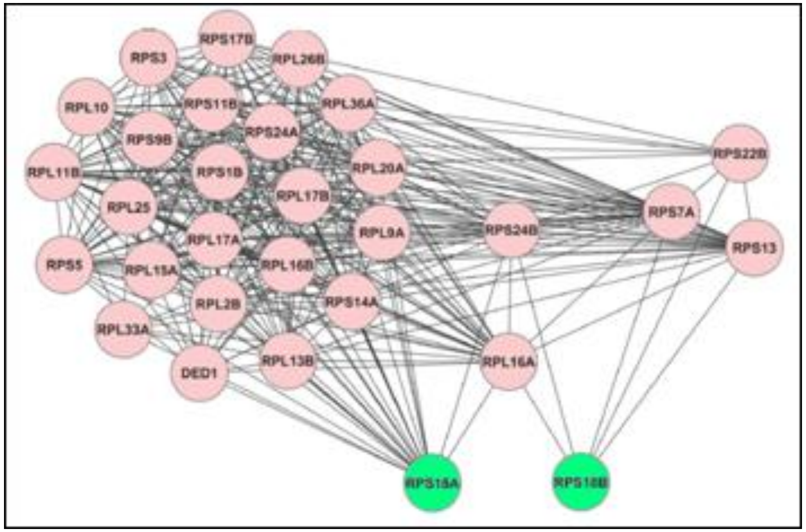
- Why?
  - ➢ Network topology: new source of biological information
  - ➢ Complementary to sequence data
  - ➢ Sequence and network topology give insight into complementary slices of biological information



Introduction

E.g.: Topology and Sequence: complementary sources of homology info.

- 59 of the yeast ribosomal proteins – retained two genomic copies
- Are duplicated proteins functionally redundant?
- No: have different genetic requirements for their assembly and localization, so are functionally distinct
- Also note: avg sequence identity of struct. similar prots ~8-10%
- E.g., two pairs with identical sequence:



100% sequence identity

50% GDV similarity

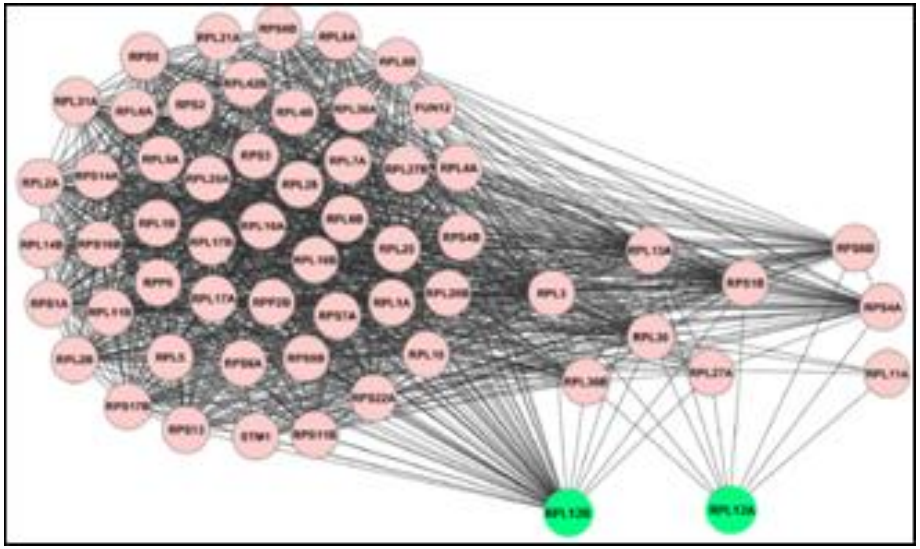
Degrees 25 and 5

V. Memisevic, T. Milenkovic and N. Przulj, “Complementarity of network and sequence information in homologous proteins”, *J. Integrative Bioinformatics*, 2010.

Introduction

E.g.: Topology and Sequence: complementary sources of homology info.

- 59 of the yeast ribosomal proteins – retained two genomic copies
- Are duplicated proteins functionally redundant?
- No: have different genetic requirements for their assembly and localization, so are functionally distinct
- Also note: avg sequence identity of struct. similar prots ~8-10%
- E.g., two pairs with identical sequence:



100% sequence identity

65% GDV similarity

Degrees 54 and 9

V. Memisevic, T. Milenkovic and N. Przulj, “Complementarity of network and sequence information in homologous proteins”, *J. Integrative Bioinformatics*, 2010.

Introduction

E.g.: Topology and Sequence: complementary sources of homology info.

⇒ Sequence and network topology
complementary slices of homology information

⇒ Redefine homology from topology?

⇒ But how?

⇒ Need network alignment algorithms.

Introduction

- We will survey computational methodology for network alignment and biological questions it may be able to answer
- Conceptually, network alignment is the process of contrasting two or more interaction networks, representing different:
  - species,
  - conditions (eg, healthy vs. disease),
  - interaction types (eg, physical vs. genetic interactions), or
  - time points

Introduction

- Based on the identified network similarities, answer a number of fundamental biological questions:
  - Which proteins, protein interactions and groups of proteins/interactions are likely to have equivalent functions across species?
  - Can we predict new functional information about proteins and interactions that are poorly characterized?
  - What do these relationships tell us about the evolution of proteins, networks, and whole species?

Introduction

- Noise in the data – screens for PPI detection report large numbers of false-positives and negatives:
    - Which interactions represent true binding events?
    - Confidence of interactions should be taken into account before network comparison
    - However
        - A false-positive interaction is unlikely to be reproduced across the interaction maps of multiple species
        - Hence, use network comparison to identify “core” interactions conserved in multiple species

Types of Network Comparisons

- Such questions have motivated 3 types (modes) of comparative methods:
  1. **Network alignment**
  2. **Network integration**
  3. **Network querying**

Types of Network Comparisons

1. Network alignment:

    - The process of comparison of two or more networks of <u>the same type</u> to identify regions of similarity and dissimilarity

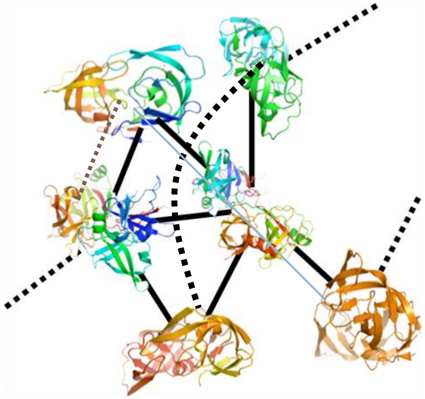
    - Commonly applied to detect subnetworks that are conserved across species and hence likely to present true functional modules

Types of Network Comparisons

2. Network integration:

- The process of combining networks encompassing interactions of different types over the same set of elements (e.g., PPI and genetic interactions) to study their interrelations

- Can assist in uncovering protein modules supported by interactions of different types



Types of Network Comparisons

- A grand challenge:

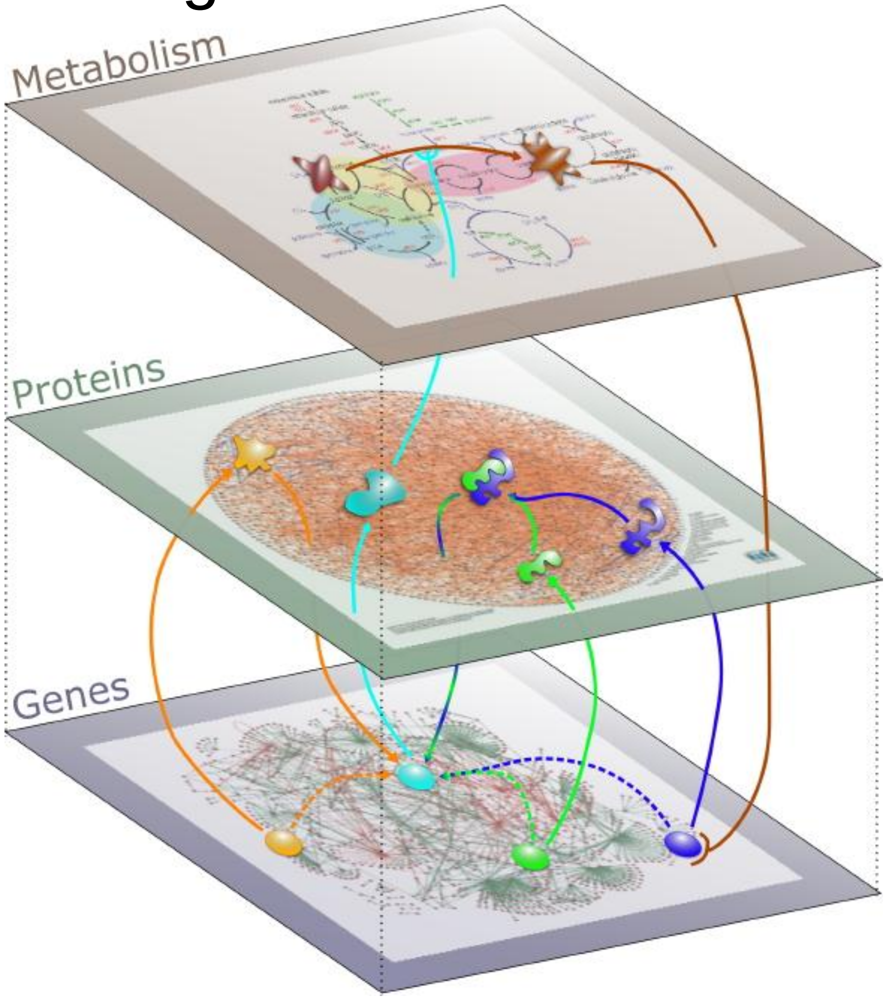


Image from: http://www-dsv.cea.fr/en/institutes/institute-of-biology-and-technology-saclay-ibitec-s/

Types of Network Comparisons

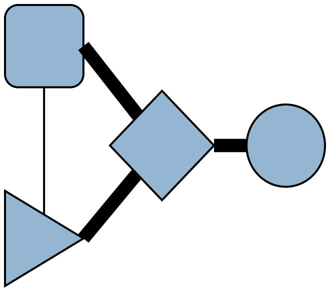
3. Network querying:

   - A given network is searched for subnetworks that are similar to a subnetwork query of interest
   - This basic database search operation is aimed at transferring biological knowledge within and across species
   - Currently limited to very sparse graphs, e.g., trees

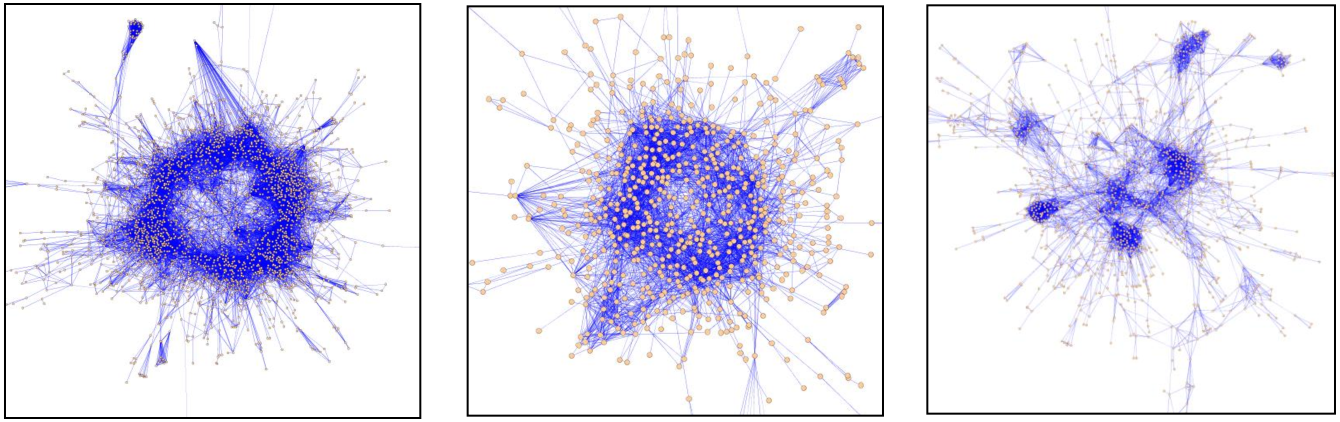
Types of Network Comparisons

3. Network querying
   - Useful application for biologists: given a candidate module, align to a database of networks ("query-to-database")

Query:



Database:



Types of Network Comparisons

Summary

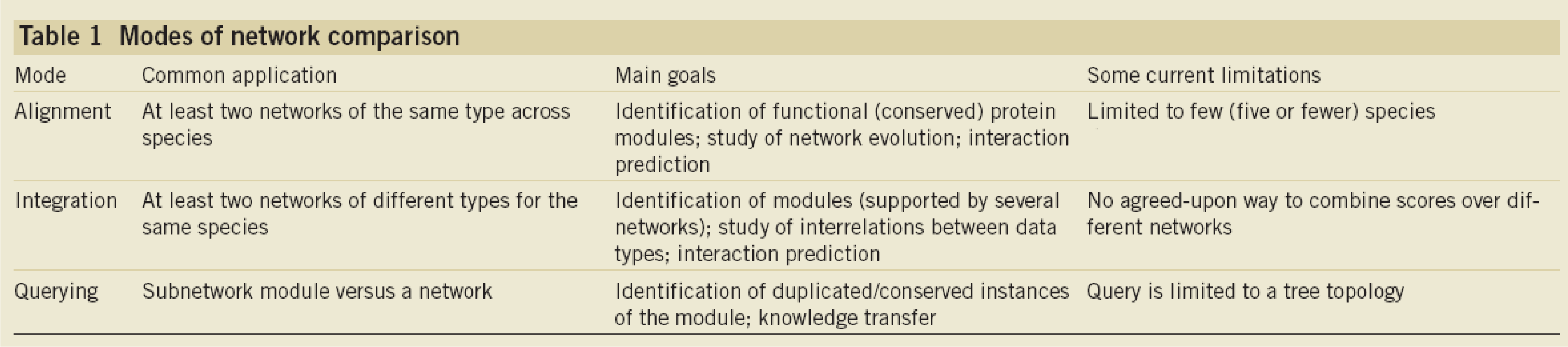


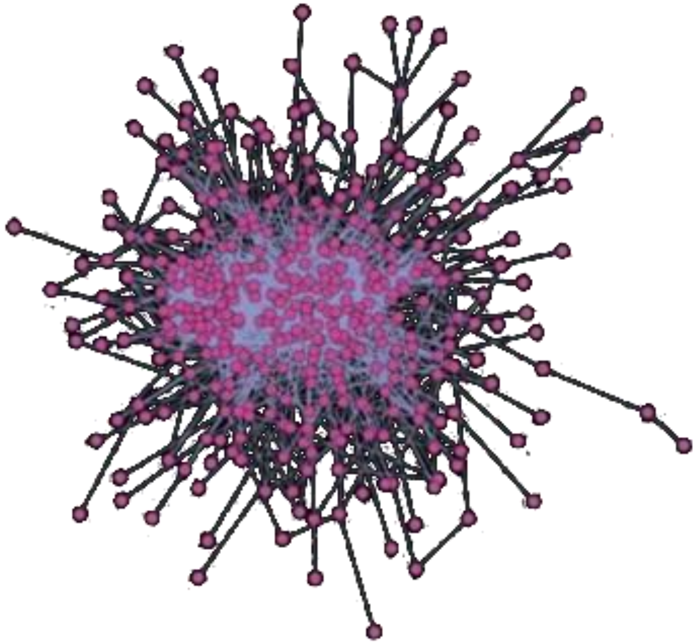
Table 1 Modes of network comparison

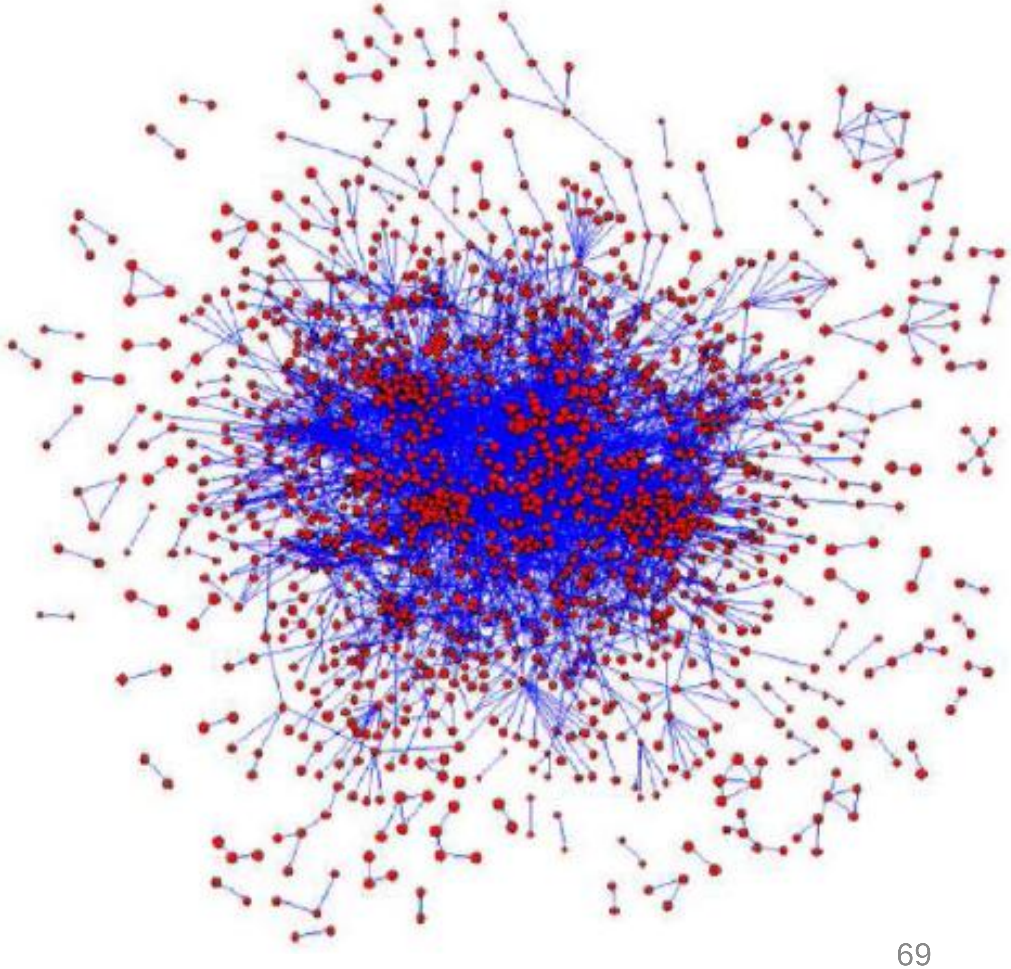
Mode	Common application	Main goals	Some current limitations
Alignment	At least two networks of the same type across species	Identification of functional (conserved) protein modules; study of network evolution; interaction prediction	Limited to few (five or fewer) species
Integration	At least two networks of different types for the same species	Identification of modules (supported by several networks); study of interrelations between data types; interaction prediction	No agreed-upon way to combine scores over different networks
Querying	Subnetwork module versus a network	Identification of duplicated/conserved instances of the module; knowledge transfer	Query is limited to a tree topology

Sharan and Ideker (2006) *Nature Biotechnology* 24(4): 427-433

Network Alignment

- Finding structural similarities between two networks



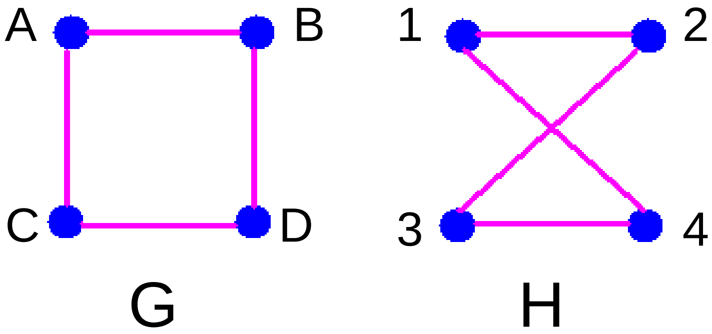


Network Alignment

Recall

Subgraph isomorphism (NP-complete):

- An isomorphism is a bijection between nodes of two networks G and H that preserves edge adjacency

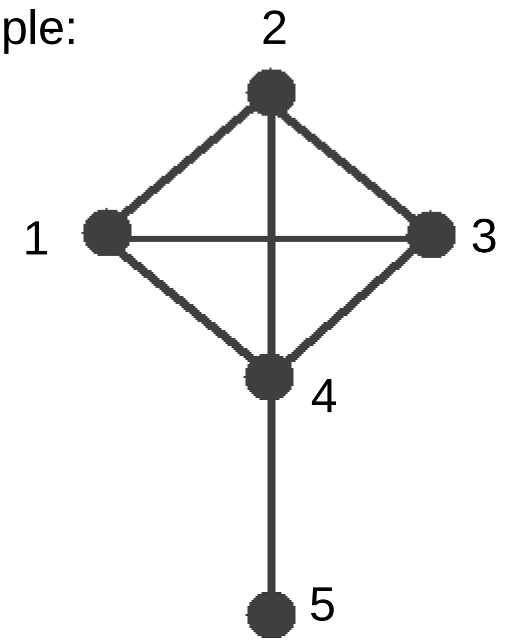


A → 1
B → 2
C → 4
D → 3

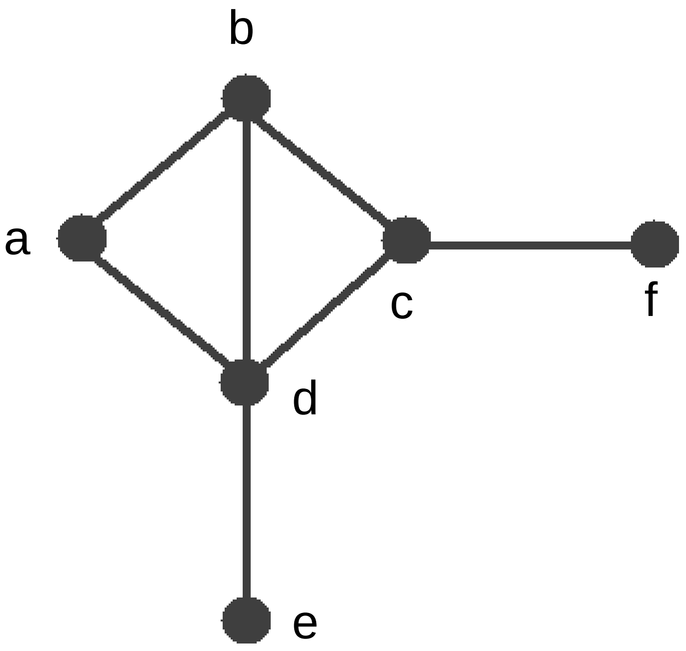
- Exact comparisons inappropriate in biology (biological variation)
- Network alignment
  - More general problem of finding the best way to “fit” G into H even if G does not exist as an exact subgraph of H

Network Alignment

Example:



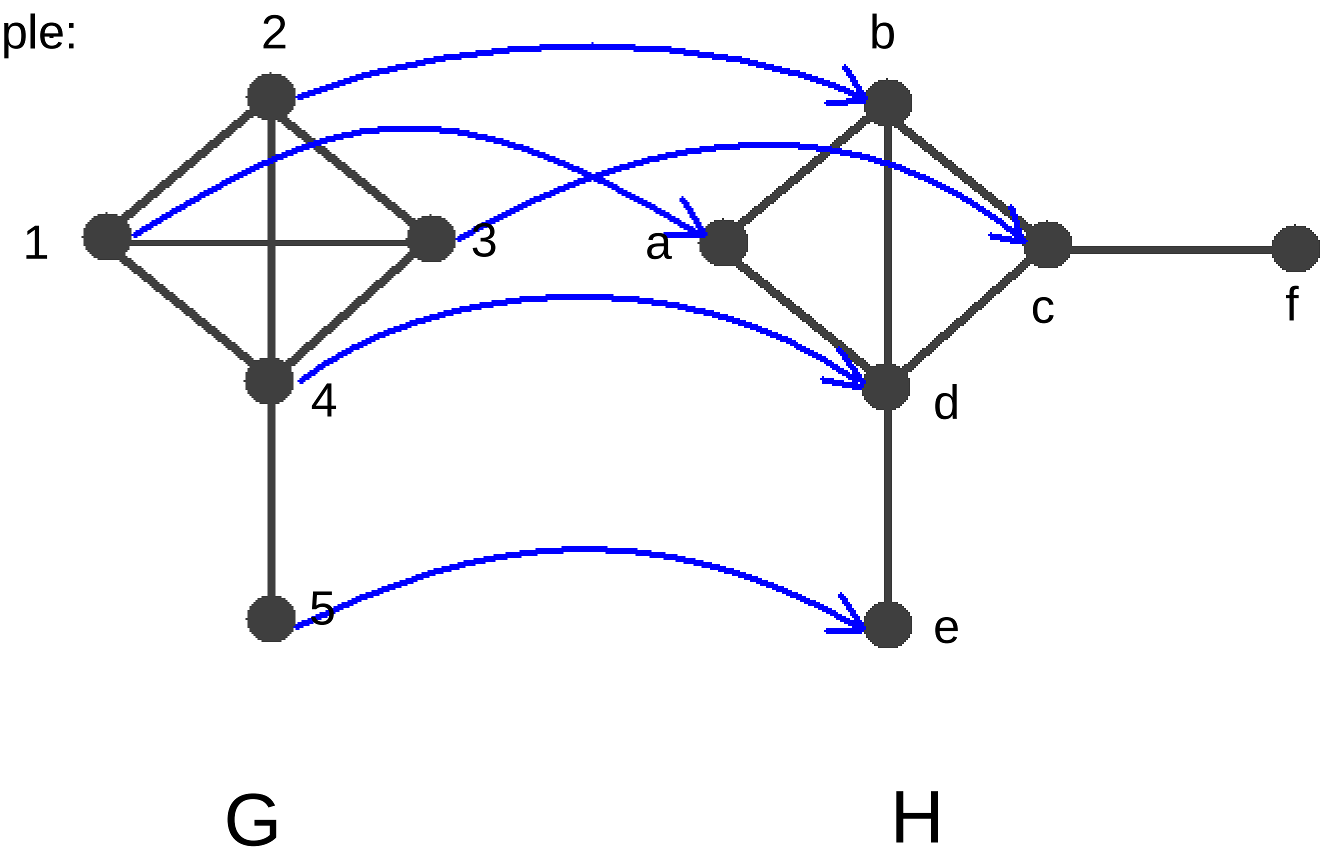
G



H

Network Alignment

Example:

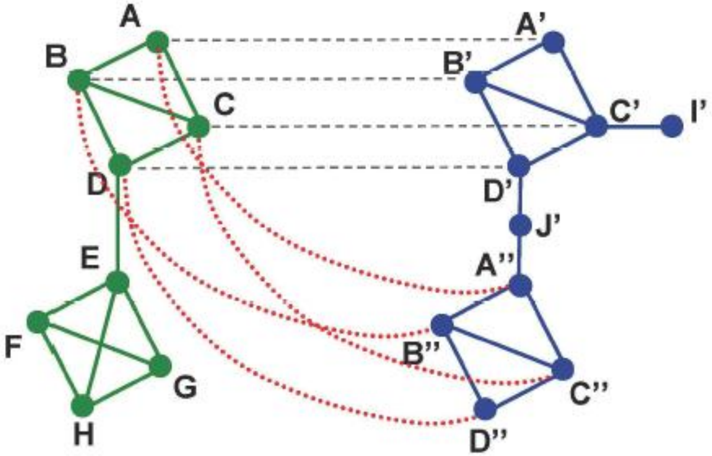


Network Alignment

- Methods vary in these aspects:
  A. Global vs. local
  B. Pairwise vs. multiple
  C. Functional vs. topological information

Network Alignment

- Methods vary in these aspects:
  A. **Global vs. local**
  B. Pairwise vs. multiple
  C. Functional vs. topological information



A. **Local alignment**:

➢ Mappings are chosen independently for each region of similarity

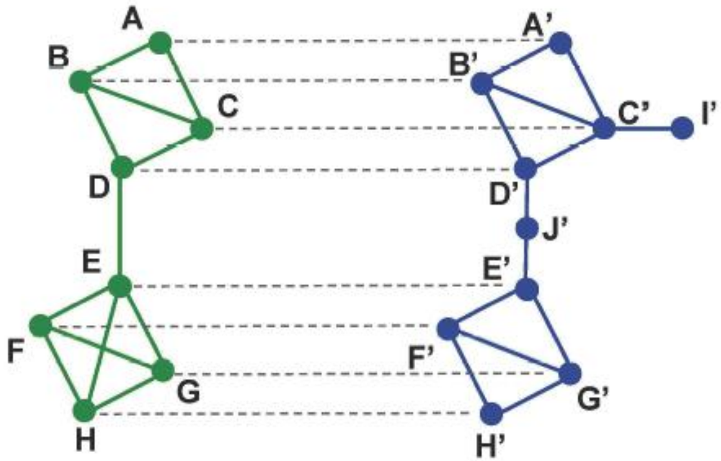
➢ Can be ambiguous, with one node having different pairings in different local alignments

➢ Example algorithms:

PathBLAST, NetworkBLAST, MaWISh, Graemlin

Network Alignment

- Methods vary in these aspects:
  A. **Global vs. local**
  B. Pairwise vs. multiple
  C. Functional vs. topological information



A. <u>Global alignment</u>:

- ➢ Provides a unique alignment from every node in the smaller network to exactly one node in the larger network
- ➢ May lead to inoptimal matchings in some local regions
- ➢ Example algorithms:

  IsoRank, IsoRankN, Graemlin 2, GRAAL, H-GRAAL

Network Alignment

- Methods vary in these aspects:
  - A. **Global vs. local**
  - B. Pairwise vs. multiple
  - C. Functional vs. topological information

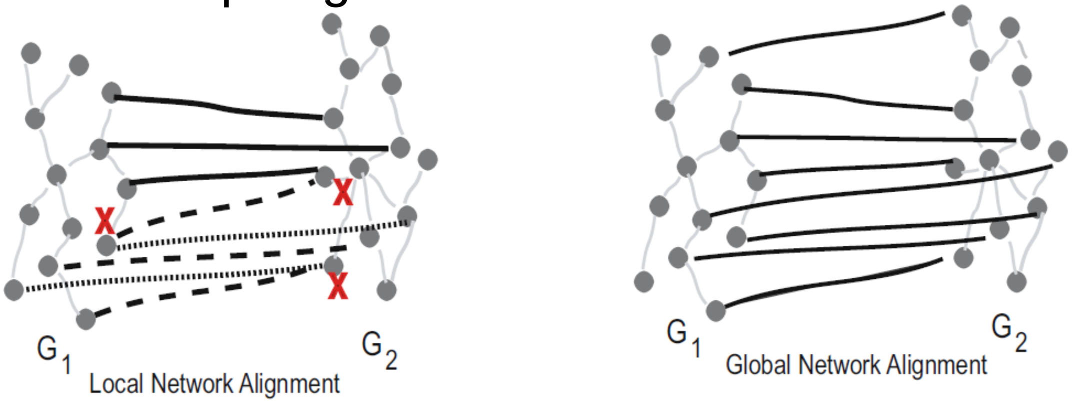
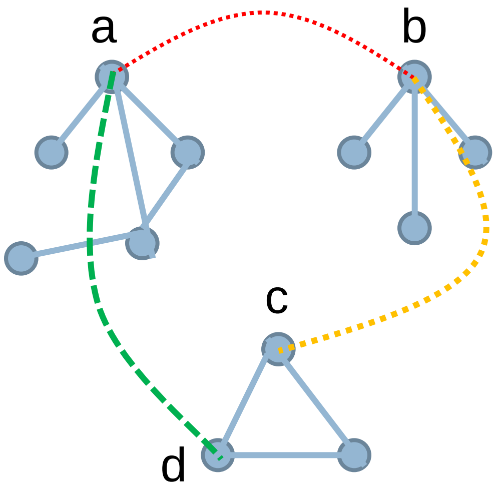


Fig. 1. Cartoon comparing global and local network alignments: The local network alignment between G_1 and G_2 specifies three different alignments; the mappings for each are marked by a different kind of line (solid, dashed, dotted). Each alignment describes a small common subgraph. Local alignments need not be consistent in their mapping— the points marked with 'X' each have ambiguous/inconsistent mappings under different alignments. In global network alignment, the maximum common subgraph is desired and it is required that the mapping for a node be unambiguous. In both cases, there are 'gap' nodes for which no mappings could be predicted (here, the nodes with no incident black edges are such nodes).

Figure taken from Singh *et al.*, RECOMB 2007, LNBI 4453, pp. 16–31, 2007.

Network Alignment

- Methods vary in these aspects:
  A. Global vs. local
  B. Pairwise vs. multiple
  C. Functional vs. topological information



B. Pairwise alignment:

- ➢ Two networks aligned
- ➢ Example algorithms:
  GRAAL, H-GRAAL, PathBLAST, MaWISh, IsoRank

Multiple alignment:

- ➢ More than two networks aligned
- ➢ Computationally more difficult than pairwise alignment
- ➢ Example algorithms:
  Greamlin, Extended PathBLAST, Extended IsoRank

Network Alignment

- Methods vary in these aspects:
  A. Global vs. local
  B. Pairwise vs. multiple
  C. **Functional vs. topological information**

C. Functional information

- ➢ Information external to network topology (e.g., protein sequence) used to define "similarity" between nodes
- ➢ Careful: mixing different biological data types, that might agree or contradict
- ➢ Example algorithms:

  all except for *GRAAL* and *H-GRAAL;* some can exclude sequence, e.g. *IsoRank*, but then perform poorly

Topological information

- ➢ Only network topology used to define node "similarity"
- ➢ Good – since it answers how much and what type of biological information can be extracted from topology only

Network Alignment

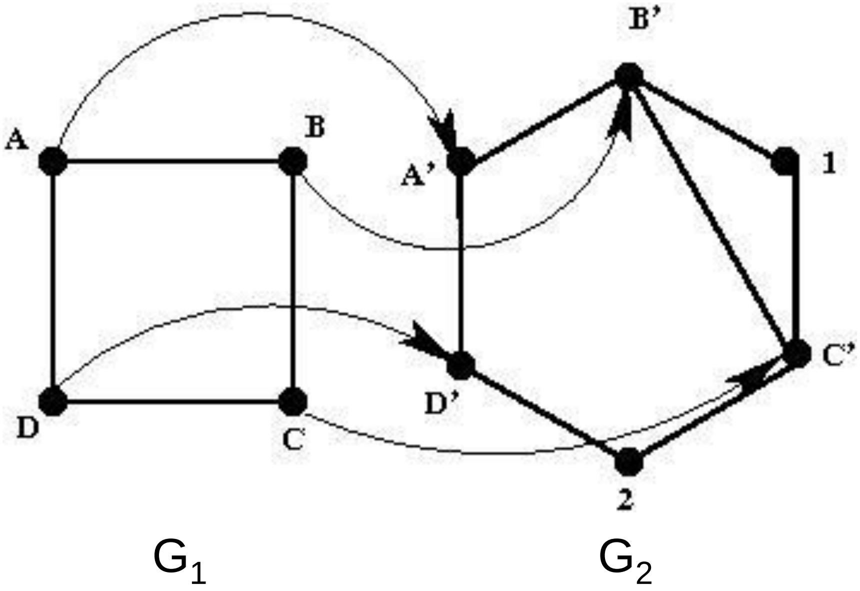
- In general, the network alignment problem is computationally hard (generalizing subgraph isomorphism)
- Hence, heuristic approaches are devised

- For now, let us assume that we have a heuristic algorithm for network alignment
- How do we measure the quality of its resulting alignments?

Network Alignment

In a network alignment, we have interaction:

- **matches (conserved interactions)** – contribute to EC
- **mismatches (gaps)**:
  1. Insertions
  2. Deletions



If G_2 evolved from G_1:

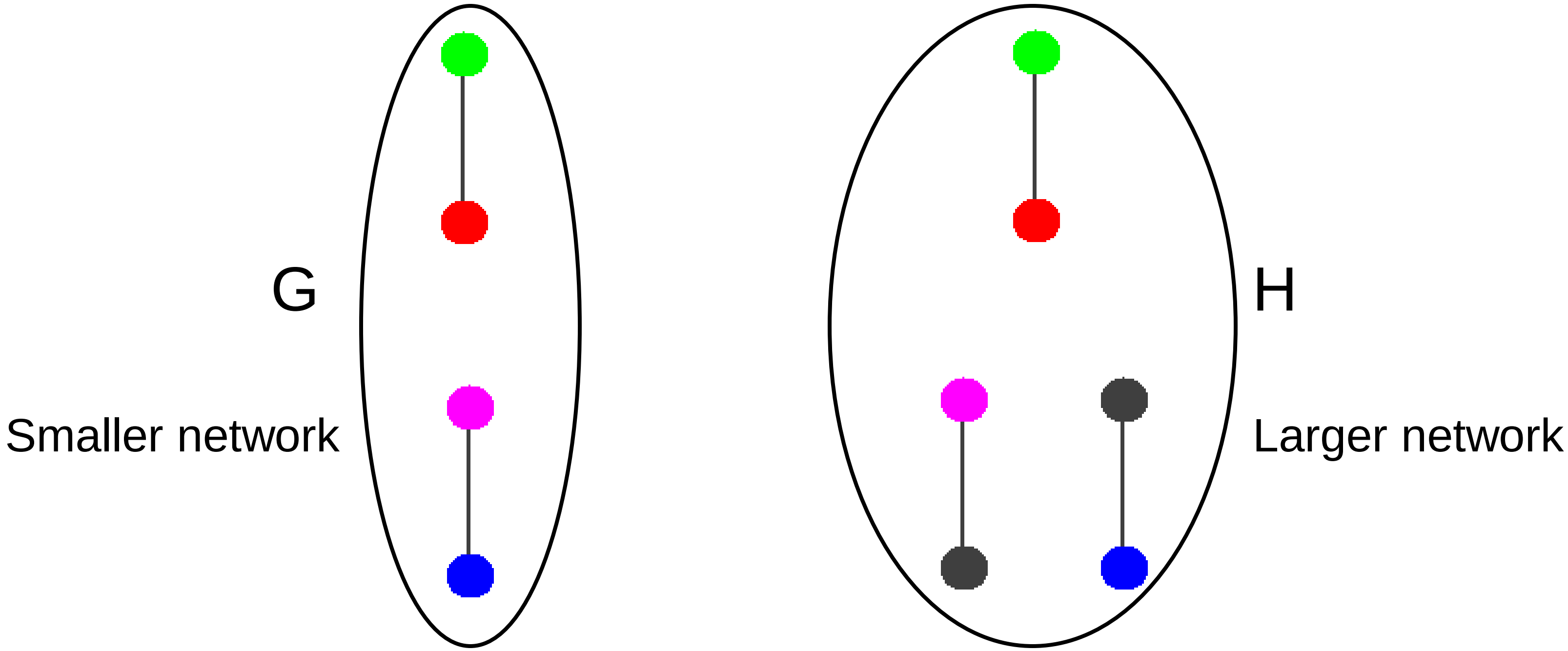
Gaps: B'1C' vs BC (**insertion** of node 1)
C'2D' vs CD (**insertion** of node 2)

If G_1 evolved from G_2:

Gaps: B'1C' vs BC (**deletion** of node 1)
C'2D' vs CD (**deletion** of node 2)

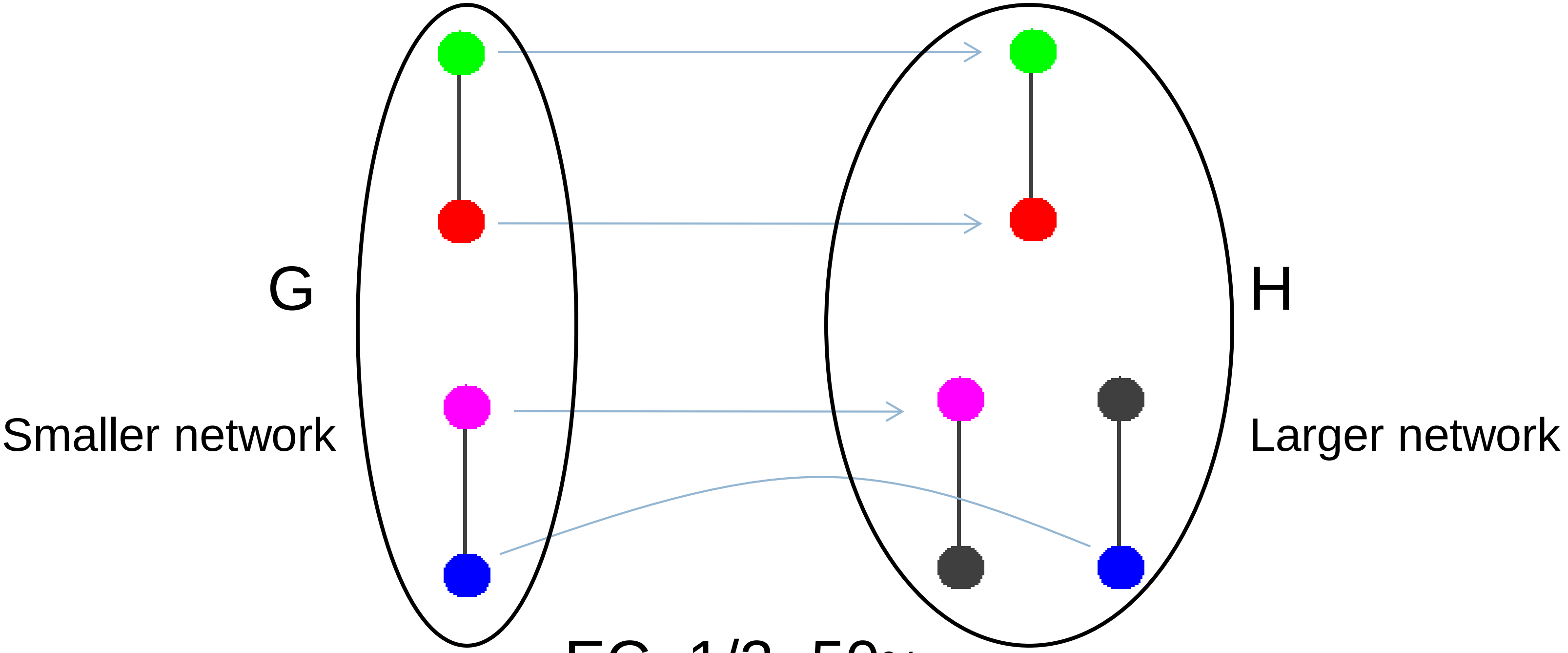
Network Alignment

- Measuring the alignment quality
  1) Edge correctness (*EC*)
     - Percentage of edges in G that are aligned to edges in H



Network Alignment

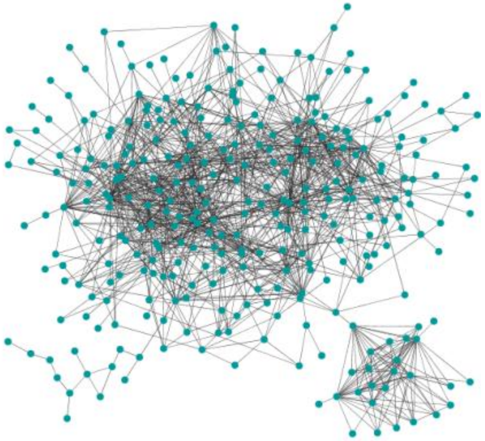
- Measuring the alignment quality
  1) Edge correctness (*EC*)
     - Percentage of edges in G that are aligned to edges in H



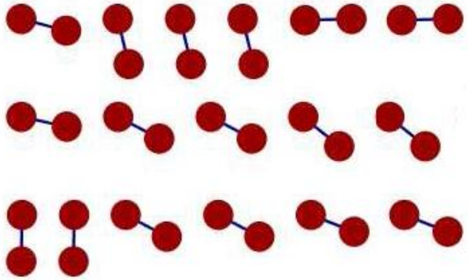
EC=1/2=50%

Network Alignment

- Measuring the alignment quality
  1) Edge correctness (*EC*)
     - Percentage of edges in G that are aligned to edges in H
  2) Size of the common connected subgraphs (CCSs)
     - Connected subgraphs (not necessarily induced) that appear in both networks
     - Is it mostly a large and contiguous alignment, or consisting of many small, disconnected fragments?



vs



Network Alignment

- Measuring the alignment quality
  3) Can the alignment be attributed to chance?
     - Compare it with a *random* alignment of the two networks
     - Compare it with the amount of alignment found between *model networks* (random graphs) of the size of the data
  4) Biological quality of the alignment:
     - Do the aligned (annotated) protein pairs have the same biological function?
     - Does the alignment identify evolutionary conserved functional modules?
     - How much of the network alignment supported by sequence alignment? **<u>Note:</u>** We should not expect networks and sequences to give identical results!!

Network Alignment

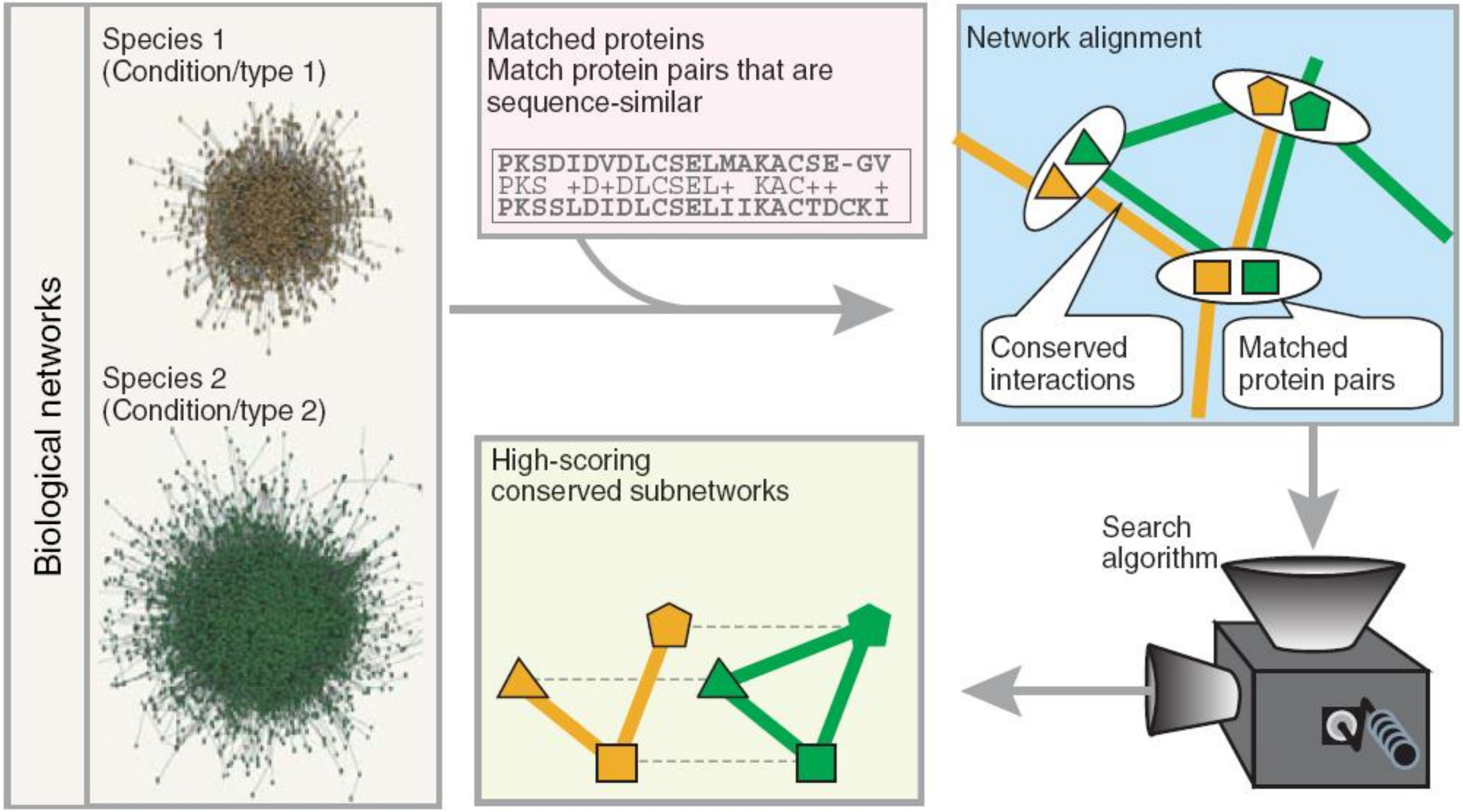
- Measuring the alignment quality
  1) Edge correctness (*EC*)
  2) Size of CCSs
  3) Statistical significance
  4) Biological quality of the alignment

Always compare your results to those of other methods

- On the same data (both synthetic and real-world data)
  - Synthetic: e.g., a PPI network with x% of rewired edges
- With respect to as many criteria as possible

Network Alignment

"Network alignment graph"



Sharan and Ideker (2006) *Nature Biotechnology* **24**(4): 427-433

Network Alignment

“Network alignment graph”

- A merged representation of the two networks being compared in which:
  - *Nodes* represent sets of molecules, one from each network
  - *Edges* represent conserved molecular interactions across different networks
- The alignment is simple when there exists a 1-to-1 correspondence between molecules across the two networks, but in general there may be a complex many-to-many correspondence
- Then apply a greedy algorithm for identifying conserved subnetworks embedded in the “network alignment graph”

Network Alignment

"Network alignment graph"

- Facilitates the search for conserved network regions
- E.g.,
  - conserved dense clusters of interactions may indicate protein complexes
  - conserved linear paths may correspond to signalling pathways
- Finding conserved pathways was done by finding "high-scoring" paths in the alignment graph (Kelley et al., *PNAS*, 2003):
  - PathBLAST
  - Identified five regions conserved across PPI networks of yeast *S. Cerevisiae* and *Helicobacter pylori*
  - Later extended to detect conserved protein clusters rather than paths (NetworkBlast)

Network Alignment

- **Key algorithmic components** of network alignment algorithms:
    - Node similarity measure
    - Rapid identification of high-scoring alignments

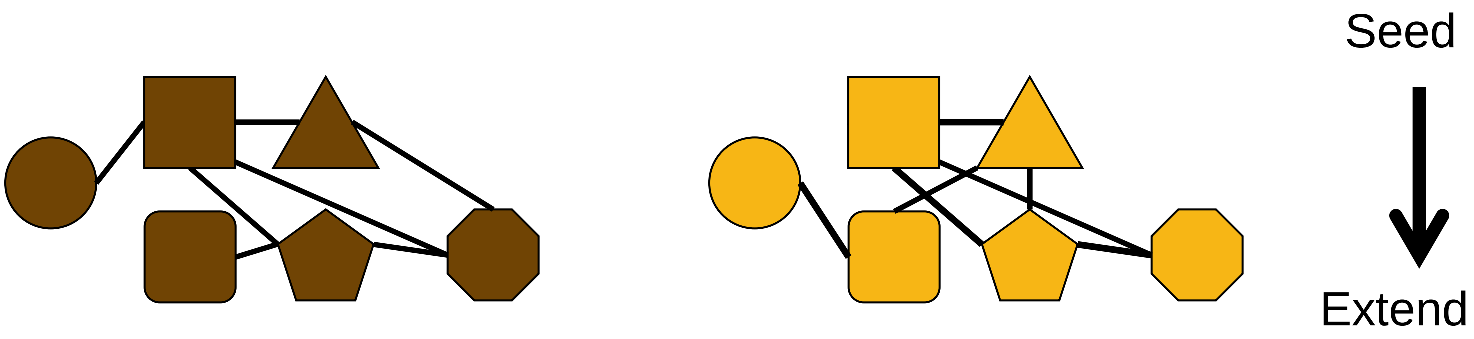
      from among the exponentially large set of possible alignments

Network Alignment

- How is **“similarity” between nodes** defined?
  - Using information external to network topology, e.g., the sequence alignment score
    - Homology, E-values, sequence similarity vs. sequence identity…
  - Using only network topology, e.g., node degree, **graphlet degree vectors** (e.g., *GRAAL, H-GRAAL*)
  - Using a combination of the two
    - But one still needs to ensure that a meaningful alignment is a result of the alignment algorithm applied to network topology, and not of the external node information
    - **Caution** about the validation/application of the algorithm
      - If sequence is used to guide the algorithm, you should not use the alignment to validate it with or make predictions about sequence-based information

Network Alignment

- Idea: seeded alignment
  - Inspired by seeded sequence alignment (BLAST)
  - Identify regions of network in which “good” alignments likely to be found
    - MaWISh does this, using high-degree nodes for seeds
    - GRAAL uses GDV similarity of nodes

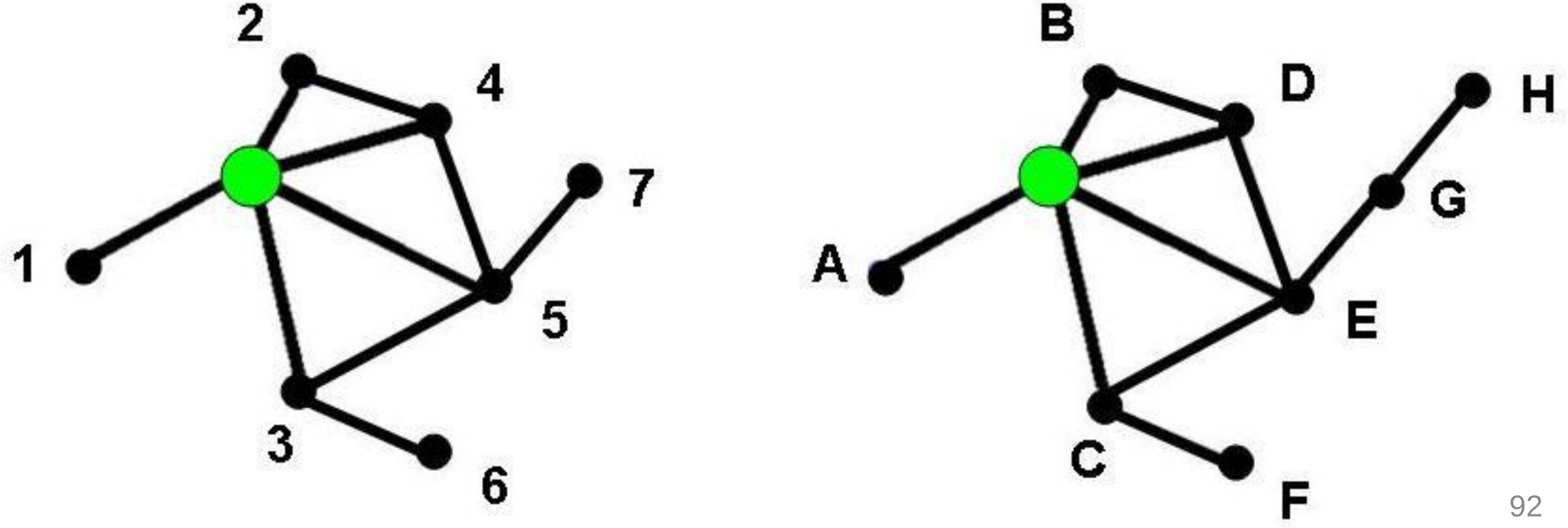


Seed

Extend

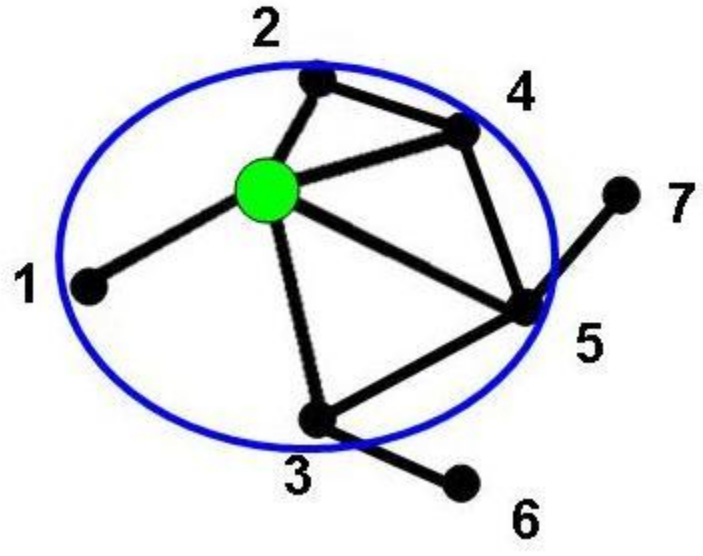
Network Alignment

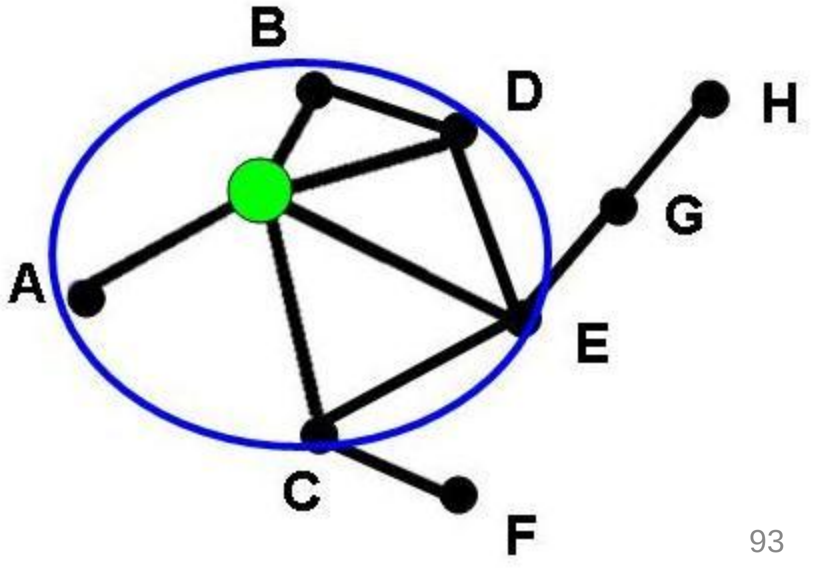
- How to identify high-scoring alignments?
  - Greedy ***seed and extend*** approaches
    - Use the most “similar” nodes across the two networks as “anchors” or **“seed nodes”**
    - “Extend around” the seed nodes in a greedy fashion



Network Alignment

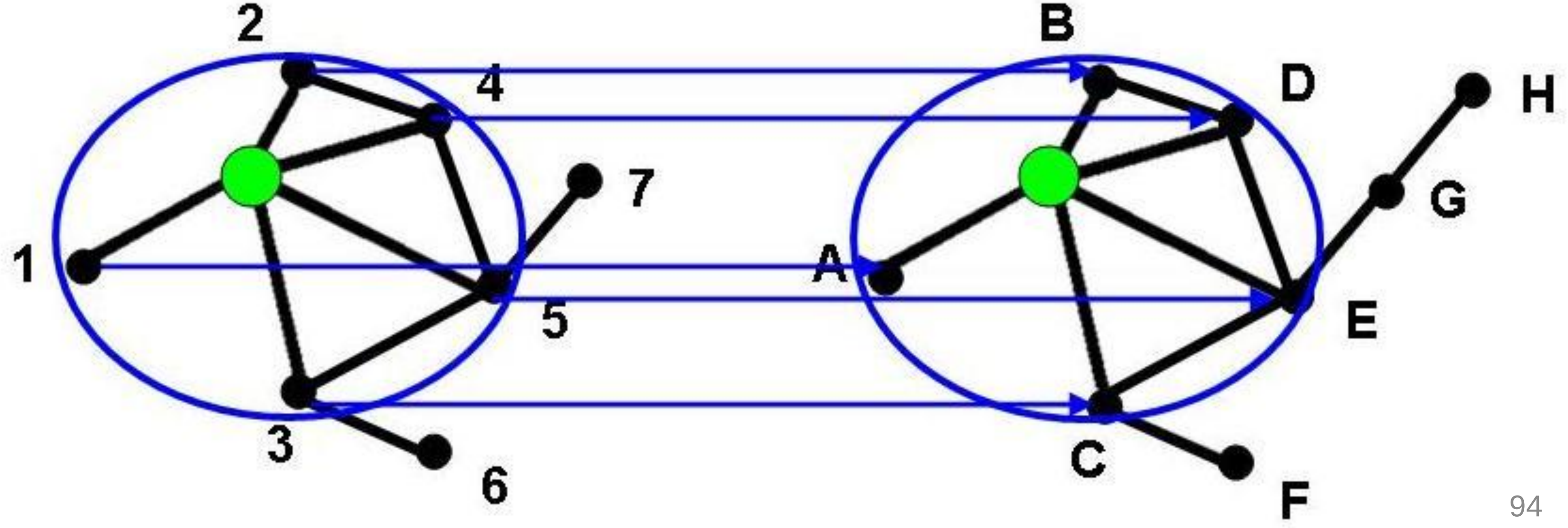
- How to identify high-scoring alignments?
  - Greedy ***seed and extend*** approaches
    - Use the most “similar” nodes across the two networks as “anchors” or **“seed nodes”**
    - “Extend around” the seed nodes in a greedy fashion





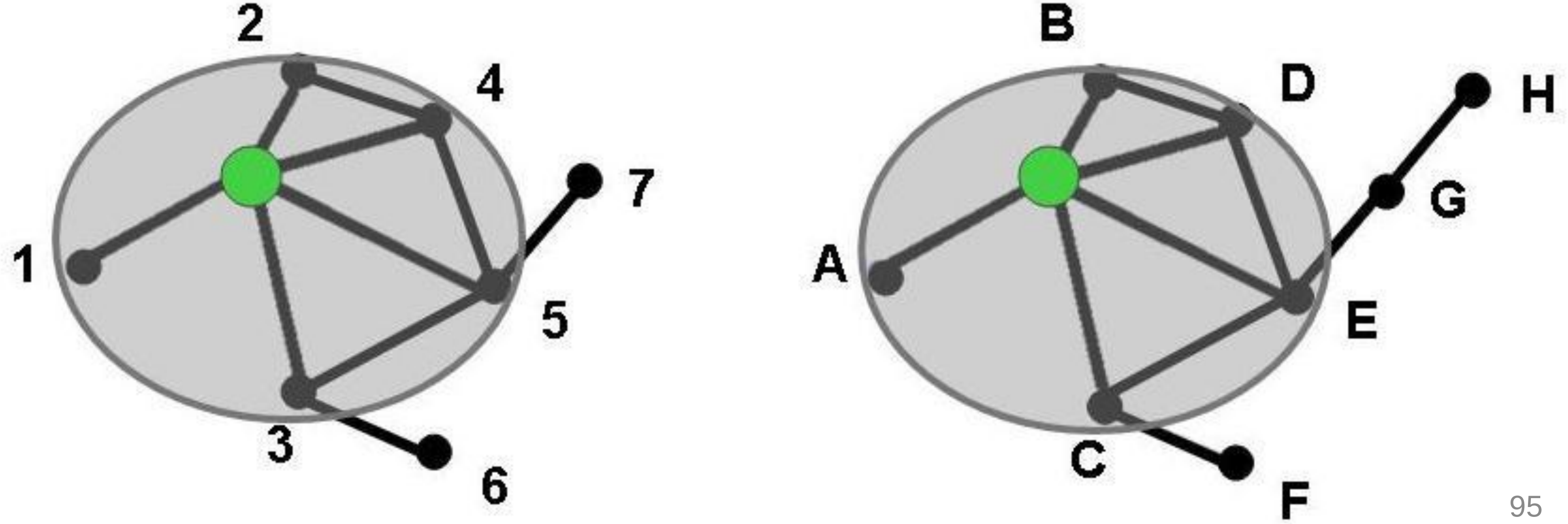
Network Alignment

- How to identify high-scoring alignments?
  - Greedy ***seed and extend*** approaches
    - Use the most “similar” nodes across the two networks as “anchors” or **“seed nodes”**
    - “Extend around” the seed nodes in a greedy fashion



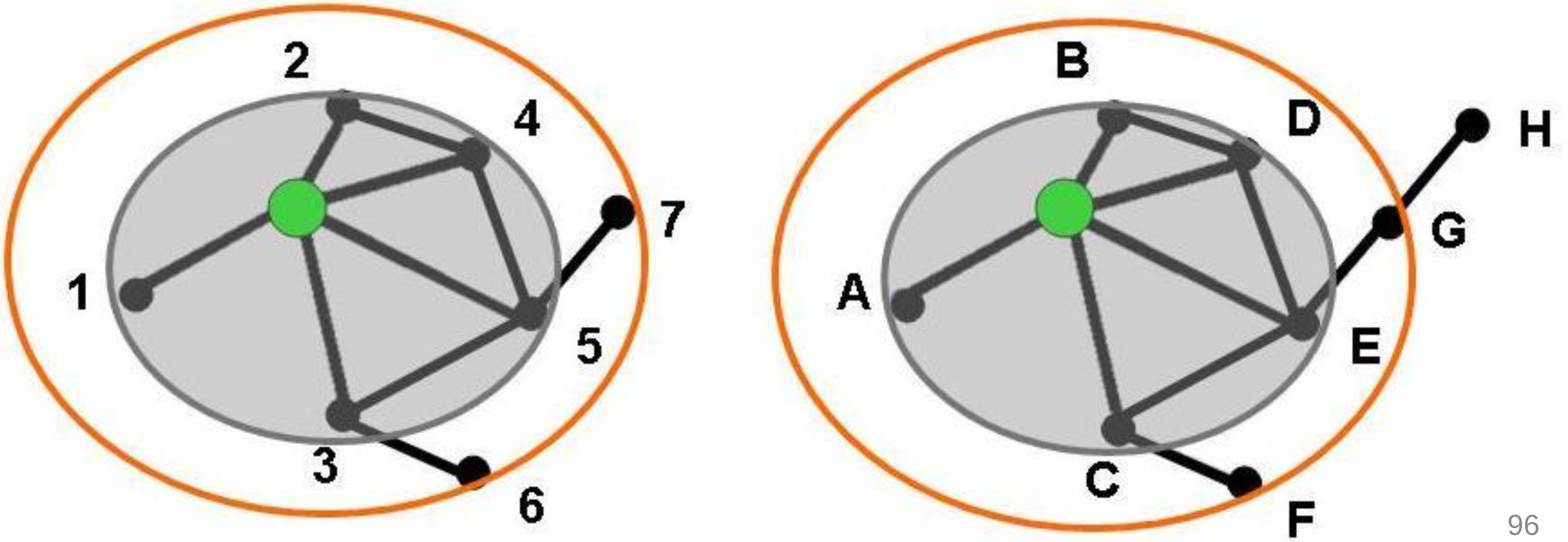
Network Alignment

- How to identify high-scoring alignments?
  - Greedy ***seed and extend*** approaches
    - Use the most "similar" nodes across the two networks as "anchors" or **"seed nodes"**
    - "Extend around" the seed nodes in a greedy fashion



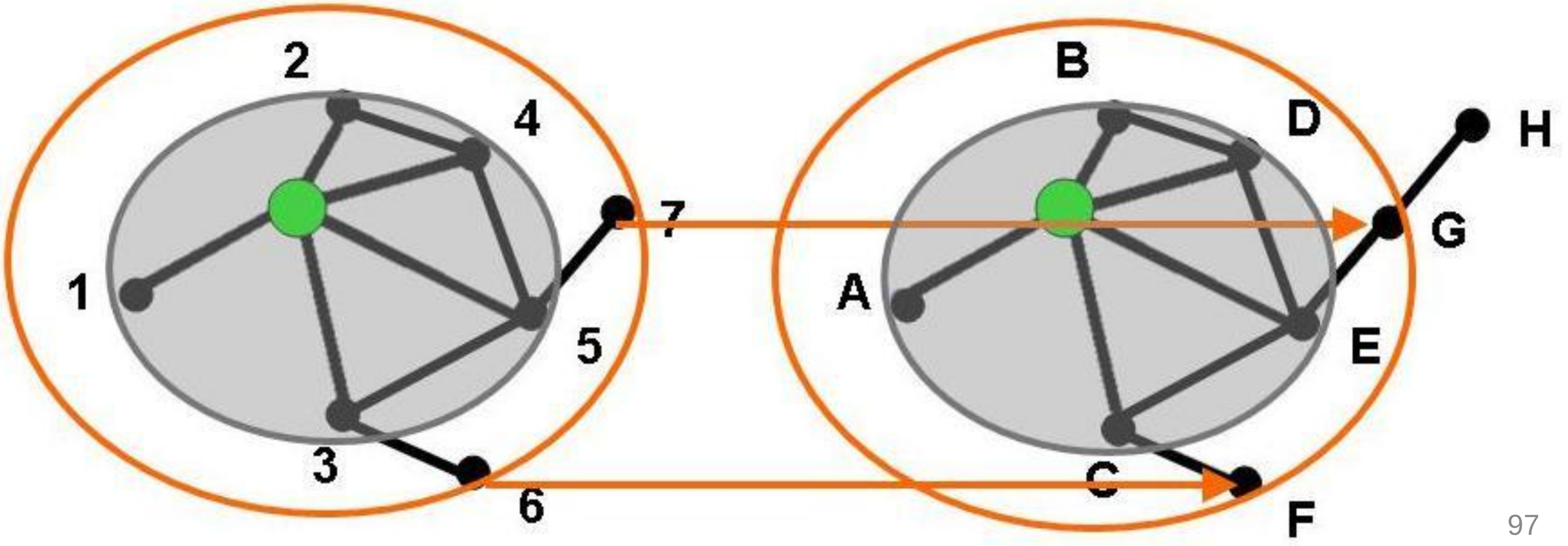
Network Alignment

- How to identify high-scoring alignments?
  - Greedy ***seed and extend*** approaches
    - Use the most “similar” nodes across the two networks as “anchors” or **“seed nodes”**
    - “Extend around” the seed nodes in a greedy fashion



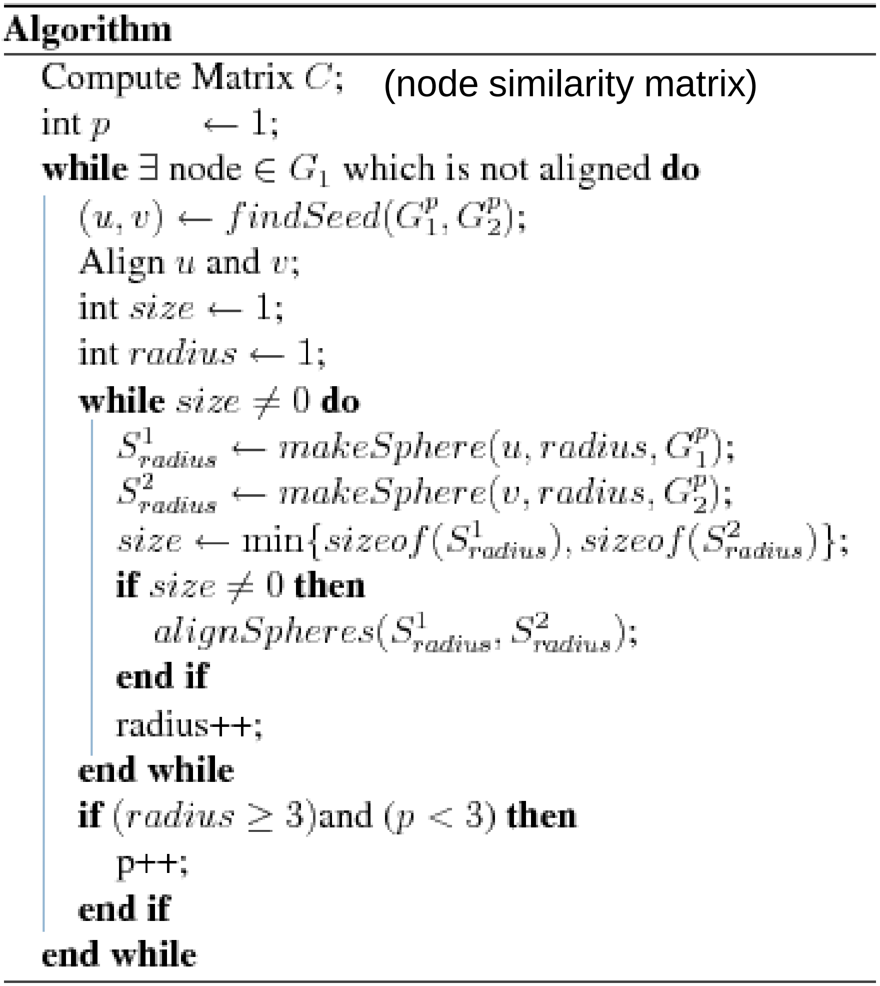
Network Alignment

- How to identify high-scoring alignments?
  - Greedy ***seed and extend*** approaches
    - Use the most “similar” nodes across the two networks as “anchors” or **“seed nodes”**
    - “Extend around” the seed nodes in a greedy fashion



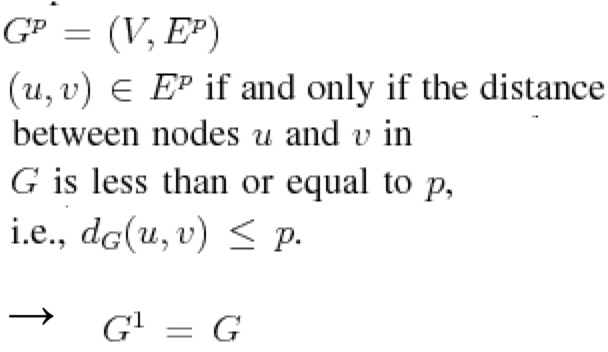
Network Alignment

- GRAAL



```
Algorithm
  Compute Matrix $C$;     (node similarity matrix)
  int $p \leftarrow 1$;
  while $\exists$ node $\in G_1$ which is not aligned do
    $(u, v) \leftarrow findSeed(G_1^p, G_2^p)$;
    Align $u$ and $v$;
    int $size \leftarrow 1$;
    int $radius \leftarrow 1$;
    while $size \neq 0$ do
      $S^1_{radius} \leftarrow makeSphere(u, radius, G_1^p)$;
      $S^2_{radius} \leftarrow makeSphere(v, radius, G_2^p)$;
      $size \leftarrow \min\{sizeof(S^1_{radius}), sizeof(S^2_{radius})\}$;
      if $size \neq 0$ then
        $alignSpheres(S^1_{radius}, S^2_{radius})$;
      end if
      radius++;
    end while
    if $(radius \geq 3)$and $(p < 3)$ then
      p++;
    end if
  end while
```

Power of graph G:



$G^p = (V, E^p)$

$(u, v) \in E^p$ if and only if the distance between nodes u and v in G is less than or equal to p, i.e., $d_G(u, v) \leq p$.

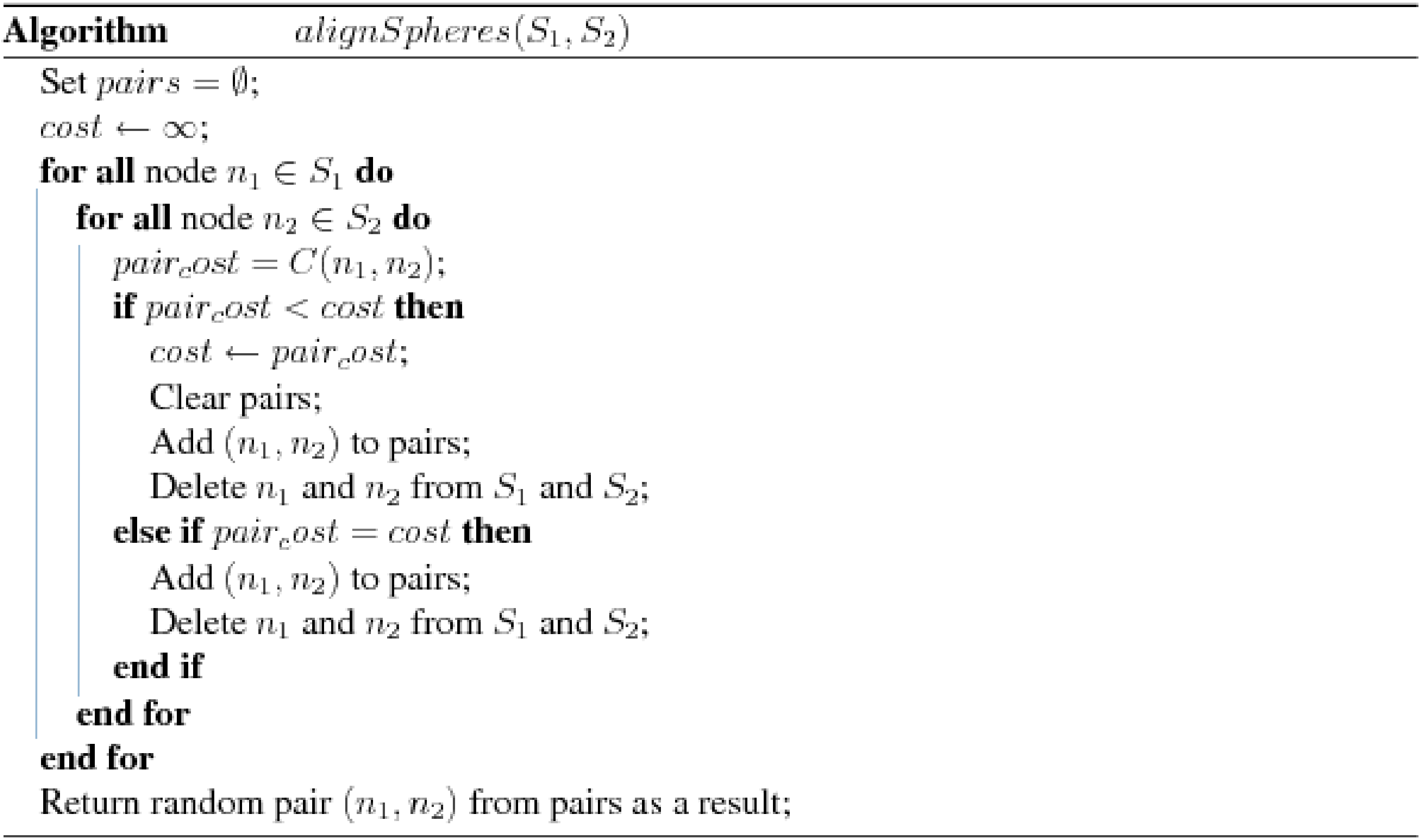
$\rightarrow \quad G^1 = G$

Code open-source:
http://bio-nets.doc.ic.ac.uk/graphcrunch2/

O. Kuchaiev, T. Milenkovic, V. Memisevic, W. Hayes and N. Przulj, “Topological Network Alignment Uncovers Biological Function and Phylogeny”, *J. Roy Soc. Interface,* 2010.

Network Alignment

- GRAAL

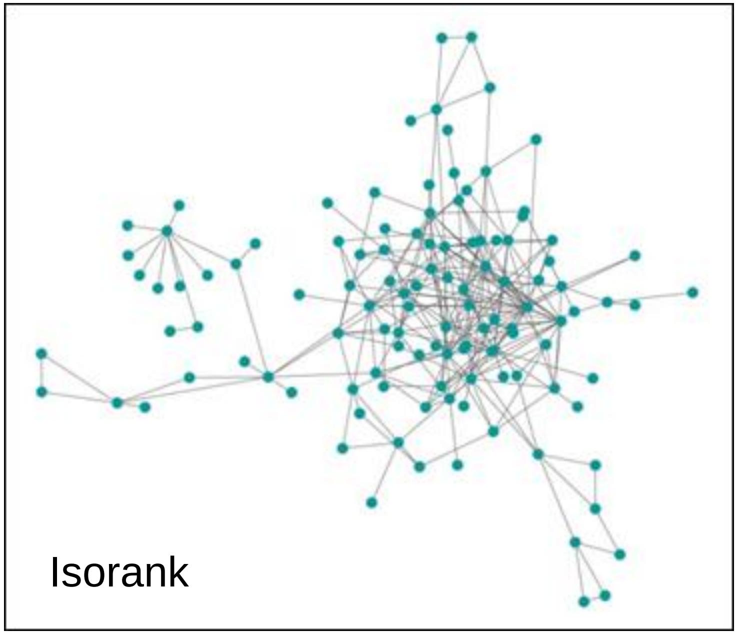


```
Algorithm    $alignSpheres(S_1, S_2)$
Set $pairs = \emptyset$;
$cost \leftarrow \infty$;
for all node $n_1 \in S_1$ do
    for all node $n_2 \in S_2$ do
        $pair_cost = C(n_1, n_2)$;
        if $pair_cost < cost$ then
            $cost \leftarrow pair_cost$;
            Clear pairs;
            Add $(n_1, n_2)$ to pairs;
            Delete $n_1$ and $n_2$ from $S_1$ and $S_2$;
        else if $pair_cost = cost$ then
            Add $(n_1, n_2)$ to pairs;
            Delete $n_1$ and $n_2$ from $S_1$ and $S_2$;
        end if
    end for
end for
Return random pair $(n_1, n_2)$ from pairs as a result;
```

O. Kuchaiev, T. Milenkovic, V. Memisevic, W. Hayes and N. Przulj, “Topological Network Alignment Uncovers Biological Function and Phylogeny”, *J. Roy Soc. Interface,* 2010.

Network Alignment

- GRAAL
- Example alignment:
  - ➢ Align PPI networks of yeast and human



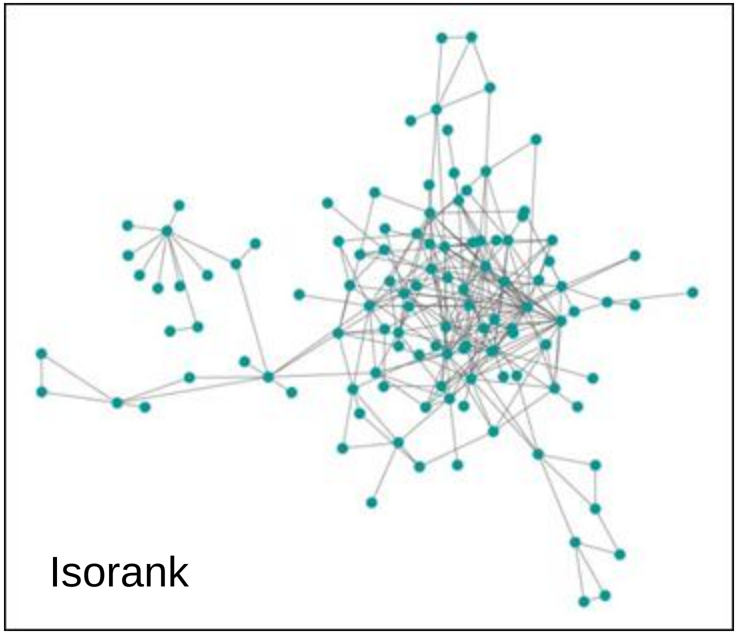
Isorank: Singh, Xu, Berger, “Pairwise Global Alignment of Protein Interaction Netowrks by Matching Neighborhood Topology,” *RECOMB* 2007, LNBI 4453, pp. 1631, 2007.

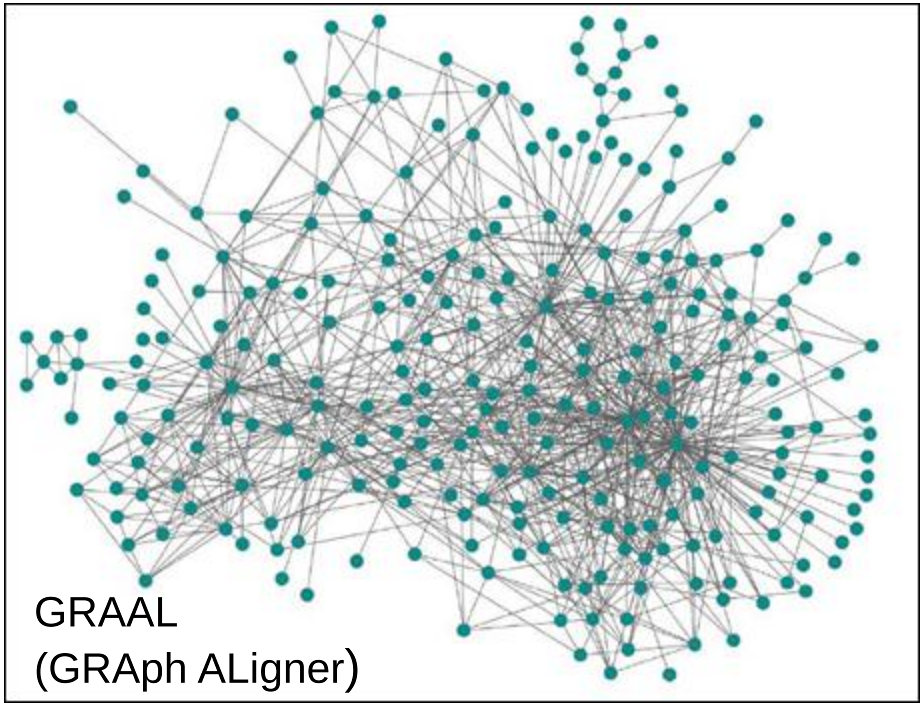
Largest CCSs

O. Kuchaiev, T. Milenkovic, V. Memisevic, W. Hayes and N. Przulj, “Topological Network Alignment Uncovers Biological Function and Phylogeny”, *J. Roy Soc. Interface,* 2010.

Network Alignment

- GRAAL
- Example alignment:
  - ➢ Align PPI networks of yeast and human





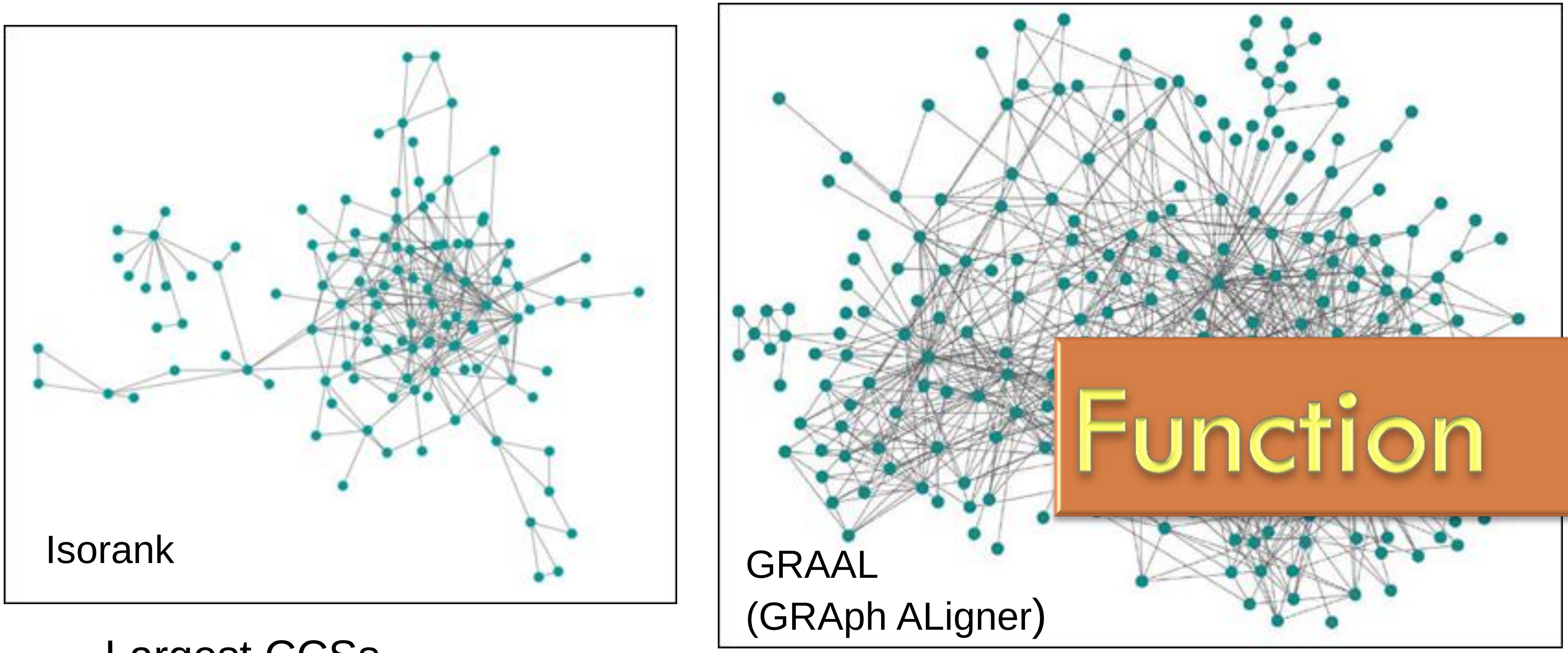
Largest CCSs

O. Kuchaiev, T. Milenkovic, V. Memisevic, W. Hayes and N. Przulj, "Topological Network Alignment Uncovers Biological Function and Phylogeny", *J. Roy Soc. Interface,* 2010.

Network Alignment

- GRAAL
- Example alignment:

  - ➢ Align PPI networks of yeast and human



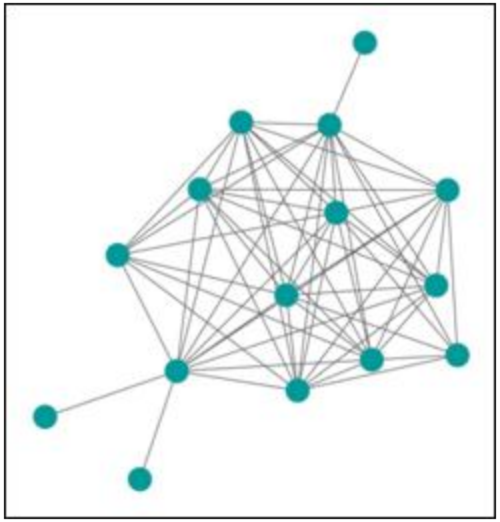
Function

Largest CCSs

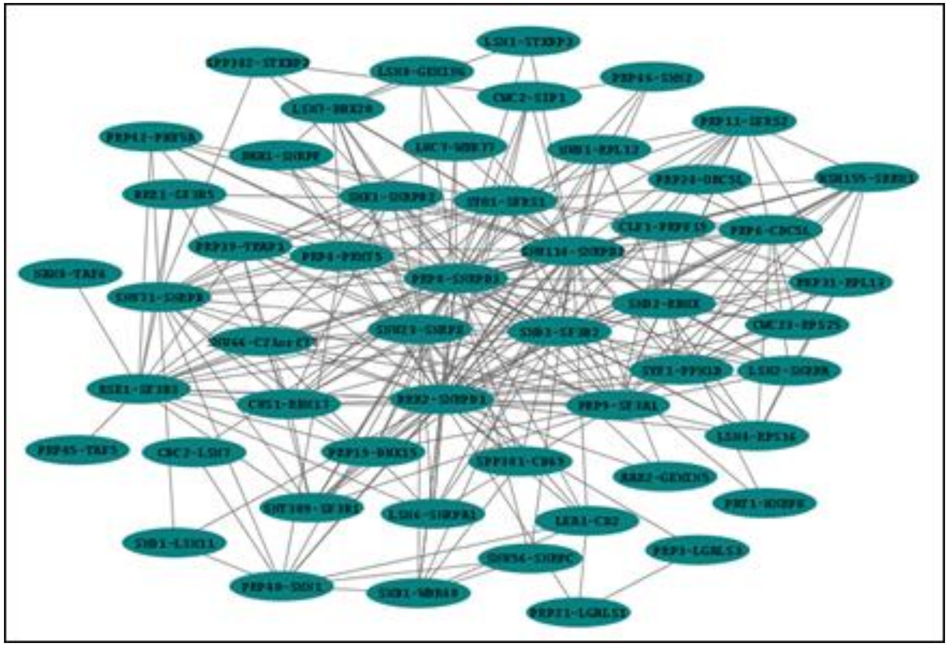
O. Kuchaiev, T. Milenkovic, V. Memisevic, W. Hayes and N. Przulj, “Topological Network Alignment Uncovers Biological Function and Phylogeny”, *J. Roy Soc. Interface,* 2010.

Network Alignment

- GRAAL
- Example alignment:
  - ➢ Align PPI networks of yeast and human



Isorank



GRAAL
(GRAph ALigner)

Second largest CCSs

O. Kuchaiev, T. Milenkovic, V. Memisevic, W. Hayes and N. Przulj, “Topological Network Alignment Uncovers Biological Function and Phylogeny”, *J. Roy Soc. Interface,* 2010.

Network Alignment

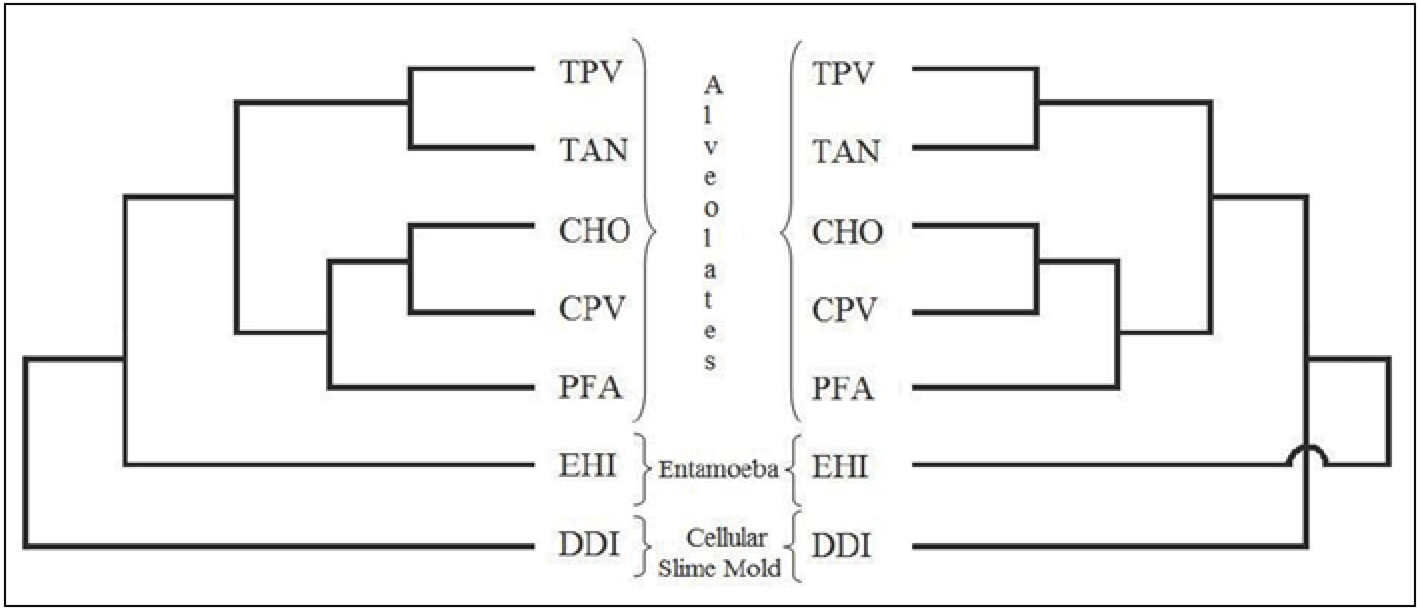
- GRAAL
- Example alignment:
    - Align metabolic networks of Protists

O. Kuchaiev, T. Milenkovic, V. Memisevic, W. Hayes and N. Przulj, “Topological Network Alignment Uncovers Biological Function and Phylogeny”, *J. Roy Soc. Interface,* 2010.

Network Alignment

- GRAAL
- Example alignment:

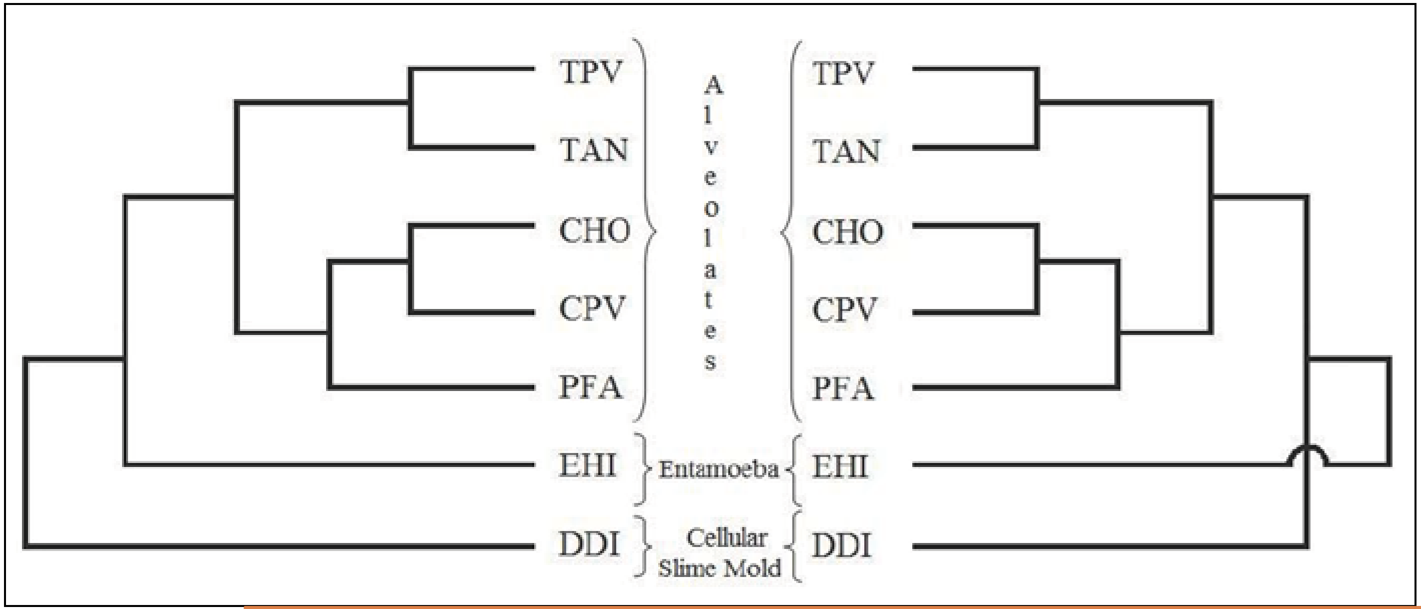
  - Align metabolic networks of Protists



O. Kuchaiev, T. Milenkovic, V. Memisevic, W. Hayes and N. Przulj, “Topological Network Alignment Uncovers Biological Function and Phylogeny”, *J. Roy Soc. Interface,* 2010.

Network Alignment

- GRAAL
- Example alignment:
  - Align metabolic networks of Protists



All statistically significant

O. Kuchaiev, T. Milenkovic, V. Memisevic, W. Hayes and N. Przulj, "Topological Network Alignment Uncovers Biological Function and Phylogeny", *J. Roy Soc. Interface,* 2010.

Network Alignment

- How to identify high-scoring alignments?
  - Greedy ***seed and extend*** approaches
    - Use the most “similar” nodes across the two networks as “anchors” or **“seed nodes”**
    - “Extend around” the seed nodes in a greedy fashion

- **GRAAL – uses GDV similarity of nodes**

- **Finds an alignment**

- Is it optimal (with respect to the cost function)?

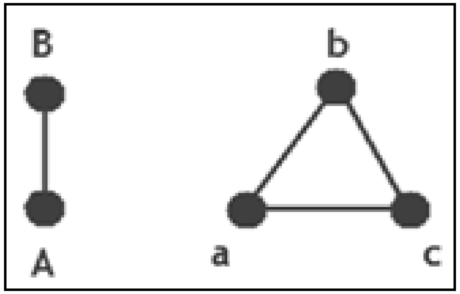
Network Alignment

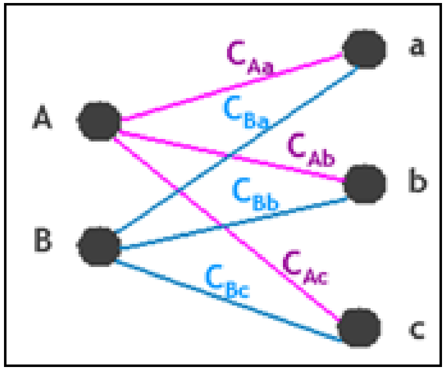
- How to identify high-scoring alignments?
  - Greedy ***seed and extend*** approaches
    - Use the most “similar” nodes across the two networks as “anchors” or **“seed nodes”**
    - “Extend around” the seed nodes in a greedy fashion

- **Alternative to matching nodes greedily based solely on node similarity scores:**
  - Align two nodes only if this increases the current alignment score
    - Reward matches (conserved interactions) – contribute to EC
    - Penalize mismatches/gaps (insertions and deletions)

Network Alignment

- How to identify high-scoring alignments?
  - Find an optimal alignment with respect to the cost function: *H-GRAAL*
    - Find GDVs of nodes across different networks
    - Align "GDV-similar" nodes, BUT not in a seed-and-extend greedy way
    - Use the *Hungarian Algorithm* for minimum weight bipartite matching
    - Hence, termed *H-GRAAL*
      - How about different optimal alignments?
      - "Core (stable) alignment" – present in all optimal alignments
      - Does not optimize EC





T. Milenkovic, W. L. Ng, W. Hayes and N. Przulj, "Optimal Network Alignment Using Graphlet Degree Vectors", *Cancer Informatics,* 9:121-137, 2010. (Highly accessed.)

Network Alignment

Weighted Bipartite Matching

Weighted bipartite matching. Given weighted bipartite graph, find maximum cardinality matching of minimum weight.

↖ m edges, n nodes

Successive shortest path algorithm. O(mn log n) time using heap-based version of Dijkstra's algorithm.

Best known bounds. $O(mn^{1/2})$ deterministic; $O(n^{2.376})$ randomized.

Planar weighted bipartite matching. $O(n^{3/2} \log^5 n)$.